ABERYSTWYTH UNIVERSITY
DOCTORATE OF PHILOSOPHY
C1250S: COMPUTATIONAL BIOLOGY

Computational recovery of enzyme haplotypes from a metagenome

Given billions of short strings of DNA 'reads' sequenced from an environment containing multiple microbial species in a structured community (many of which are unknown and most are unculturable), can we recover the set of distinct co-occurring variants which exist within the population encoding enzymes responsible for catalysing "interesting" reactions in the microbiome?

Author
Samuel Nicholls msn

Supervisors
Dr. Amanda Clare afc
Dr. Christopher Creevey chc30
Dr. Wayne Aubrey waa2

Examiners
Prof. James McInerney
University of Nottingham

Dr. Martin Swain
Aberystwyth University

This thesis is submitted in candidature for a Doctorate of Philosophy in Computational Biology (C1250S)

Final Copy
2018

Declaration

In signing below, I confirm that:

- This work has not previously been accepted in substance for any degree, and is not concurrently submitted in candidature for any other degree.
- This work is the result of my own investigations, except where otherwise stated.
- Sources are acknowledged with explicit references. A full bibliography is appended.
- This thesis has not been altered by correction or editing services.
- I have read and understand the sections on unfair practice in the Students' Examinations Handbook and the relevant sections of the current Student Handbook, and agree to abide by the University's regulations governing these issues.
- As per the *Regulations for the Submission and Examination of Research Theses*, this work was defended *viva voce*, and was approved by the Examining Board, subject to minor corrections. The agreed amendments have been completed and accepted by the internal examiner.

	Monday 30th July 2018
Signature	**Date**

Consent to Share Work

I hereby give consent for my thesis to be made available for photocopying and inter-library loan, and for the title and abstract to be made available to external organisations.

	Monday 30th July 2018
Signature	**Date**

For Clem

Although you are not here to see this,
it is because of you that I am.

Acknowledgements

So, this is it – quite possibly the largest document I will ever write alone. But science is not conducted in a vacuum, and I owe many thanks for those who have helped me to get here.

First of all, I would like to thank my primary supervisors: Amanda Clare and Christopher Creevey. Over the past three-and-a-bit years, your guidance has allowed me to develop as a scientist in my own right. You gave me the freedom to find my own way, but were always around if I needed direction. Providing a balance of both computer science, and biological sciences, I could not have asked for a better team to guide me, as the first PhD student to be joint between our two institutes.

Amanda, you've taken the brave job of supervising me twice. Thank you for allowing me not to take things too seriously. You know how to get the best out of me, and I am a better scientist because of you. Chris, thank you for taking the risk in recruiting a computer scientist looking to defect to biology, my time in your lab has widened my interest and understanding of evolutionary biology, despite your unhealthy interest in tree diagrams. I wish you and your new lab well in Belfast.

To my secondary supervisor, Wayne Aubrey, thank you for taking the time to introduce me to the art of bench science. I have learned that there are many different types of water, that 1 ml is a gigantic volume and transferring tiny volumes of colourless liquids between tiny tubes without making a mistake takes a lot of time, effort and skill. Your time, patience and constant reminding about laboratory sterile technique, has allowed me to truly become an interdisciplinary scientist.

To my colleagues in KU Leuven, and specifically Kurt de Grave and Leander Schietgat; thank you for hosting me during the first summer of my PhD adventure, and for your suggestions in formulating the mathematical grounding for the metahaplome.

To my unofficial mentor, Arwyn Edwards, thank you for taking me under your wing, inducting me to the Nanopore cabal, and for your help in debugging my grants and fellowship applications via satellite phone from a glacier.

To my friends old and new, my family, and my partner Sam; I have sacrificed many occasions to conduct lab work, debug code, and write this book. I hope that looking back, it is worth it. Thank you all for your long-term patience, confidence, and support. I am greatly looking forward to getting my life back, and spending it with you.

To my old lab colleagues Tom and Francesco, thank you for your support during the start of my PhD. You made me feel welcome to both the lab, and to the study of biology. I've missed sharing a rant about our cluster over a pint, we'll have to catch up soon. And to my fellow remaining PhD candidates; Jess, Ben and Nick, good luck. I am going to a better place.

To my "boyf" Tom, thank you for all the procrastination-based adventures. Your friendship has been adequate, and now that this is done, we can design, build and break more things, sequence more Monsters and build more lab tat. I am proud that we enable good science done cheap.

To Sammy; my late-night lab-partner, my eagle-eyed proof-reader, my best friend. Thank you for picking me up and carrying me through the last mile, you have borne the brunt of my stresses and I cannot thank you enough. Writing this work has been a mammoth effort, and it would not have been possible without your encouragement, patience and optimism.

* * *

Submitting this thesis now draws my time at Aberystwyth to a close. I have spent the majority of my adult life here in this beautiful seaside town. I have become an expert in seagull dodging, hill walking and even managed to enjoy the occasionally sunny day where is has been warm enough to warrant a visit to the beach. To my home department of Computer Science, thank you for your support over the past eight years, and for letting me use the coffee room as an office for the past three.
I will miss you, and this place.

Finally, to you, my examiners, and perhaps optimistically, my future readers. This book is the sum of my life's work so far. Since my late grandfather introduced me to computing over twenty years ago, I have always had the aspiration to become a scientist. Submitting this thesis is a realisation of that aspiration, and a significant milestone in the adventure of a computer scientist who found that biology posed far more interesting problems to solve, and is a field in which it is more acceptable to wear a white lab coat. I want to thank you, for taking the time to read through this work, and I sincerely hope that you enjoy reading it as much as I endured writing it.

Abstract

Population-level diversity of microbial communities (microbiomes) represent a biotechnological resource for biomining, biorefining and synthetic biology; but industrial exploitation of enzymes responsible for catalyzing reactions of interest requires the recovery of the exact DNA sequences (or "haplotypes") that encode the genes. However, haplotype reconstruction is an extremely difficult computational problem, further complicated by the infancy of techniques for the handling of environmental sequencing data (metagenomics). Current haplotyping approaches cannot choose between alternative haplotype reconstructions and fail to provide biological evidence of correct predictions. Additionally, there is no philosophical framework under which we can consider the variation of genes within a microbial community, such as those that encode isoforms of enzymes of interest to us.

To address this, my thesis proposes the "metahaplome" as a definition for the set of haplotypes for a genomic region of interest within a microbial community. This work will offer the first formalisation of the problem of recovering haplotypes from a metagenomic data set, and present `Hansel` and `Gretel`: a novel probabilistic framework that reconstructs the most likely haplotypes from complex microbiomes. The framework is robust to sequencing error and uses all available evidence from aligned reads, without altering or discarding observed variation.

The approach is verified with multiple *in silico* experiments, including two *de facto* data sets that are currently used to benchmark algorithms for the recovery of viral quasispecies, and strain identification. With long-read sequencing, this thesis will demonstrate *in vitro* verification of the approach, presenting the first biologically validated method for the recovery of haplotypes from a microbial community. Finally, I will introduce the "Rumen Landscape" pilot study to demonstrate the sort of research questions and novel biological insight that can be obtained through exploration of the metahaplome.

Words 58,452
Version `1b58e5f`

Contents

List of Figures **xvii**

List of Tables **xxi**

List of Listings **xxiii**

Glossary **xxv**

Acronyms **xxvii**

1 Introduction **1**

1.1 The Microbiome, Hologenomes and Pangenomes 2
1.2 The Microbiome, a source of 'interesting' enzymes 5
1.3 Metagenomics, as an insight into microbial communities 8
1.4 Beyond the metagenome 9
1.5 Thesis Overview 12
1.5.1 Terminology 14
1.5.2 Contributions 16
1.5.3 Chapter Descriptions 17

2 A Review of Haplotype Assembly **19**

2.1 Introduction 20
2.2 Origins of the problem 21
2.2.1 Minimum fragment (MFR) and minimum SNP (MSR) removal 23
2.2.2 Longest haplotype reconstruction (LHR) 24
2.3 Minimum error correction (MEC) 25
2.3.1 FastHare (2004) 25
2.3.2 The first diploid genome sequence of an individual human (2007) 26
2.3.3 SpeedHap (2007) 28
2.3.4 HASH (2008) 29
2.3.5 HapCUT (2008) 31

2.3.6 Discussion . . . 32
2.4 Modern Fragment Removal . . . 32
2.4.1 Maximum Fragments Cut (MFC) and ReFHap (2010) . . . 33
2.4.2 Minimum weighted edge removal (MWER) and HapCompass (2012) . . . 34
2.4.3 Balanced Optimal Partitioning (BOP) and H-BOP (2012) . . . 36
2.5 Modern probabilistic approaches . . . 37
2.5.1 The first probabilistic formulation (Li et al. 2004) . . . 37
2.5.2 The first Markov chain model (Wang et al. 2006) . . . 40
2.5.3 The first probabilistically reconstructed diploid sequence (2007) . . . 41
2.5.4 MixSIH (2013) . . . 44
2.5.5 ProbHap (2014) . . . 46
2.5.6 ParticleHap (2015) . . . 48
2.5.7 HapCUT2 (2017) . . . 50
2.5.8 HapChat (Preprint, 2018) . . . 52
2.5.9 Discussion . . . 55
2.6 Toward polyploid-capable methods . . . 56
2.6.1 HapCompass with polyploid support (2013) . . . 56
2.6.2 HapTree (2014) . . . 58
2.6.3 Lens (2016) . . . 61
2.6.4 Discussion . . . 62
2.7 Viral quasispecies reconstruction (VQSR) . . . 63
2.7.1 ShoRAH (2011) . . . 63
2.7.2 EVORhA (2015) . . . 65
2.7.3 SAVAGE (2017) . . . 67
2.7.4 PEHaplo (Preprint, 2018) . . . 70
2.7.5 Virus-VG (Preprint, 2018) . . . 70
2.7.6 Discussion . . . 71
2.8 Reconstruction of strains from a metagenome . . . 72
2.8.1 Introduction . . . 72
2.8.2 ConStrains (2015) . . . 73
2.8.3 StrainPhlAn (2017) . . . 74
2.8.4 DESMAN (2017) . . . 75
2.9 Conclusion . . . 79

3 Theoretical Work **81**
3.1 The Metahaplome . . . 81
3.1.1 Introduction . . . 83
3.1.2 A Formal Definition . . . 85
3.1.3 Methodology . . . 87

3.1.4 Assembly and pseudo-references . . . 87
3.1.5 Summary . . . 88
3.2 Hansel: A data structure for the storage of sequence variance . . . 90
3.2.1 Different from the typical SNP matrix . . . 91
3.2.2 Different from a graph . . . 92
3.2.3 Using information from non-adjacent SNPs, and the path so far . . . 92
3.2.4 A dynamic structure . . . 94
3.2.5 Hansel as a graphical model . . . 94
3.2.6 Probabilistic edge weights . . . 95
3.2.7 Simplification of conditional edge weights . . . 96
3.2.8 Estimation of probabilities . . . 97
3.2.9 Smoothing . . . 100
3.2.10 Summary . . . 101
3.3 Gretel: An algorithm for the recovery of haplotypes from metagenomes . . . 101
3.3.1 Greedy path construction . . . 104
3.3.2 Gretel's outputs . . . 104
3.3.3 Reweighting to find multiple haplotypes . . . 104
3.3.4 Stopping criterion . . . 105
3.3.5 Haplotype scoring . . . 105
3.3.6 Summary . . . 107

4 *In silico* Evaluation 109
4.1 *In silico* Testing Overview . . . 109
4.1.1 Read generation and variant calling . . . 112
4.1.2 Evaluating recovery accuracy . . . 114
4.2 Experiment 1: Synthetic Sequences via Simulated Evolution . . . 115
4.2.1 Method . . . 115
4.2.2 Results . . . 120
4.2.3 Conclusions . . . 121
4.3 Experiment 2: Metahaplomes from real DHFR genes . . . 122
4.3.1 Method . . . 123
4.3.2 Results . . . 126
4.3.3 Initial alignments . . . 128
4.3.4 Conclusions . . . 129
4.4 Experiment 3: Recovering haplotypes from an HIV metahaplome . . . 130
4.4.1 Method . . . 132
4.4.2 Results . . . 133
4.4.3 Conclusions . . . 134
4.5 Experiment 4: Recovering haplotypes from a benchmark mock community . . . 136

4.5.1 Method . . . 137
4.5.2 Results . . . 140
4.5.3 Conclusions . . . 140

5 Recovery of enzyme haplotypes from the rumen microbiome 143
5.1 Methodology . . . 145
5.1.1 Existing Data . . . 145
5.1.2 Filtering candidate regions . . . 145
5.1.3 Final selection and primer pair design . . . 147
5.1.4 Reverse Transcription and PCR . . . 148
5.2 Results . . . 150
5.2.1 Sanger Sequencing . . . 150
5.2.2 High-Fidelity Amplicons . . . 153
5.3 Oxford Nanopore sequencing . . . 155
5.4 Haplotype Verification . . . 157
5.4.1 Post-sequencing analysis . . . 157
5.4.2 Haplotype Verification . . . 160
5.5 Conclusion . . . 160

6 A Rumen Haplotype Landscape 163
6.1 Introduction . . . 163
6.2 Methods . . . 164
6.2.1 Existing Data . . . 164
6.2.2 Alignment, Variant Calling, GFF Filtering and Haplotyping . . . 164
6.2.3 Post-processing of haplotypes . . . 166
6.2.4 Calculation of dN/dS ratios . . . 167
6.3 Results . . . 168
6.3.1 Variability between species and strains . . . 168
6.3.2 Influence of likelihood and coverage on dN/dS . . . 170
6.3.3 Strain-level rates of variation between functional categories . . . 172
6.3.4 Characterising the variability within pangenomes . . . 176
6.3.5 Characterising the strain-level variability between gene families . . . 178
6.3.6 Characterising the variability within a single protein . . . 182
6.4 Conclusion . . . 186

7 General Discussion and Conclusions 187
7.1 Performance and tractability . . . 187
7.2 Methodological comparison of our approach . . . 189
7.3 Hansel and Gretel: Future Work . . . 191
7.3.1 Reweighting . . . 191

7.3.2 Naive insertion handling . . . 192
7.3.3 Greedy Search . . . 192
7.3.4 Stopping Criterion . . . 193
7.3.5 Unused Evidence . . . 193
7.3.6 Improvements to `Hansel`'s memory footprint . . . 194
7.3.7 Likelihood normalisation . . . 194
7.4 Conclusion . . . 195

Bibliography **199**

Appendix A Terminology and Techniques for Metagenomic Data **211**
A.1 Biases . . . 212
A.2 DNA Sequencing . . . 213
A.2.1 Chain Termination Sequencing . . . 213
A.2.2 Cyclic Reversible Termination Sequencing by Synthesis . . . 214
A.2.3 Zero-Mode Waveguide Sequencing . . . 215
A.2.4 Nanopore Strand Sequencing . . . 216
A.3 Data Handling . . . 218
A.3.1 Common File Formats . . . 218
A.4 Assembly . . . 223
A.4.1 Implementations . . . 225
A.5 Alignment . . . 227
A.6 Variant Calling . . . 228

Appendix B Laboratory Notebook and Protocols **229**
B.1 Initial PCR with GoTaq . . . 230
B.2 Gel Extraction . . . 234
B.3 Sanger Sequencing . . . 236
B.4 Amplification with High Fidelity Polymerase . . . 238
B.5 Oxford Nanopore Sequencing . . . 244
B.5.1 Amplicon Pooling . . . 244
B.5.2 SQK-LSK108 Sequencing Preparation . . . 246

Appendix C Additional Circos Plots **251**
C.0.1 G31 . . . 252
C.0.2 G90 . . . 253
C.0.3 G251 . . . 254

Colophon **257**

List of Figures

1.1 A diagram contextualising the metahaplome and how it fits in the topography of the metagenome and pangenome . . . 11

2.1 An example SNP matrix . . . 21
2.2 A trivial example showing how an incorrect base call in the SNP matrix causes 'conflict' under the MFR and MSR frameworks . . . 23

3.1 Three corresponding representations of variation observed on sequencing reads, illustrating the `Hansel` matrix . . . 93
3.2 A depiction of how, by considering only adjacent SNP evidence, one may recover sequences of SNPs that are not actually supported by the reads . . . 93
3.3 A detailed example demonstrating the relationship between sequenced reads and entries in the `Hansel` matrix . . . 98
3.4 Calculation of the two edge weights corresponding to the example in Figure 3.3. . . . 99
3.5 Selection of optimum next variant given edge weight calculations from Figure 3.4. . 99

4.1 An illustration describing the intuition of `shredder` . . . 112
4.2 An illustration describing the intuition of `snpper` . . . 113
4.3 An illustration describing the concept of Hamming distance . . . 114
4.4 Hierarchy of 6300 read sets generated with `seq-gen` for the evaluation of `Gretel` . . 117
4.5 Flow diagram describing the computational pipeline for the generation and analysis of synthetic metahaplomes . . . 119
4.6 Boxplots summarising the accuracy of haplotypes recovered by `Gretel` for the 6300 synthetic metahaplomes . . . 121
4.7 Diagram demonstrating the intuition behind the *DHFR* experiment . . . 124
4.8 Depiction of synthetic reads aligned against the chosen pseudo-reference . . . 125
4.9 Proportion of SNPs correctly discovered by `snpper` for the synthetic metahaplomes generated from real *DHFR* genes . . . 125
4.10 Boxplots summarising the accuracy of haplotypes recovered by `Gretel` on read sets generated from real *DHFR* genes . . . 127

4.12 Heatmap of average coverage for the five input *DHFR* haplotypes, for two different `bowtie2` parameter choices . . . 128
4.13 `BLAST` bitscores against `Gretel` likelihoods for the recovered *HIV-1* haplotypes . . . 135
4.14 Flow diagram describing the process of conducting *in silico* testing with the Quince *et al.* mock community . . . 139
4.15 Boxplots summarising the accuracy of haplotypes recovered by `Gretel` from a synthetic microbiome . . . 141

5.1 Flow diagram depicting the computational pipeline used to select regions of interest from the GFF, align reads to the reference, recover corresponding haplotypes with `Gretel` and generate suitable primers. . . . 146
5.2 Depiction of the "flattening" method used to generate a consensus on which to search for valid primer pairs . . . 147
5.3 Heatmap describing the density of Illumina reads that mapped to the 10 selected `Gretel` candidates, for each sampled timepoint . . . 149
5.4 First gel electrophoresis result from amplification of the ten chosen Gretel candidates 149
5.5 Gel electrophoresis result after conducting PCR with an annealing gradient . . . 149
5.6 Partial chromatogram plotting Sanger sequencing signals for one of the `Gretel` candidates . . . 151
5.7 Flow diagram depicting the protocol for gene-specific reverse transcription and high-fidelity PCR . . . 152
5.8 Gel electrophoresis banding to confirm high-fidelity PCR runs . . . 153
5.9 Flow diagram depicting the pipeline to generate amplicons for Nanopore sequencing 154
5.10 Quality scores of Nanopore reads generated from sequencing of the Gretel amplicons 155
5.11 Circos plot demonstrating Sanger and Nanopore sequencing support for the haplotypes recovered from a real microbial community by `Gretel` . . . 161

6.1 Boxplot of dN/dS ratios for the 49,908 "landscape" haplotype returning regions, by species . . . 169
6.2 Boxplot of dN/dS ratios for the 49,908 "landscape" haplotype returning regions, by strain . . . 169
6.3 Boxplot of average haplotype likelihood, for the 49,908 "landscape" haplotype returning regions, by strain and sample . . . 171
6.4 Boxplot of average read coverage, for the 49,908 "landscape" haplotype returning regions, by strain and sample . . . 171
6.5 Landscape of dN/dS ratios for Information Storage and Processing categories, by strain 173
6.6 Landscape of dN/dS ratios for Cellular Processes and Signalling categories, by strain 174
6.7 Landscape of dN/dS ratios for Metabolism categories, by strain . . . 175

6.8 Jitter plot mapping average dN/dS variation (y-axis) over sets of haplotypes assigned to non-supervised orthologous group (NOGs), against the proportion of the 41 strains (x-axis) on which they appear. 176
6.9 Boxplot describing the distribution of dN/dS ratios for each pangenome within the landscape pilot 177
6.10 Comparison of the core and accessory genomes of the eight species investigated for the landscape, against *R. flavefaciens* 179
6.11 Landscape of dN/dS ratios for the 10 most abundant "core" NOGs on *R. flavefaciens*, for all strains 180
6.12 Landscape of dN/dS ratios for the 10 most abundant "accessory" NOGs on *R. flavefaciens*, for all strains 181
6.13 Barplot of windowed dN/dS ratios for a G123-like (endo-1,4-beta-xylanase) protein in the landscape pilot 183
6.14 Multiple sequence alignment for G123-like sequences recovered from *Prevotella ruminicola AR32* 184
6.15 Barplot of windowed dN/dS ratios for a G90-like protein (alpha-N-arabinofuranosidase) in the landscape pilot 185

A.1 My Oxford Nanopore Technologies `MinION` DNA sequencer, used for verification of my metagenomic haplotyping framework in Chapter 5 216
A.2 A graphical depiction of how ASCII characters encode sequence quality in `FASTQ` files. 220
A.3 A visualisation demonstrating the fundamental difference between assembly of a single organism and a metagenomic sample. 224

C.1 G31 Haplotype Circos Plot 252
C.2 G90 Haplotype Circos Plot 253
C.3 G251 Haplotype Circos Plot 254

List of Tables

4.1 Summary of data sets used for *in silico* evaluation of `Gretel` . . . 111
4.2 Comparison of mutation rate, and mean number of variants discovered by `snpper` for the synthetic read sets generated by `shredder` . . . 120
4.3 The chosen 'pseudo-reference' and five genes that constitute the synthetic *DHFR* metahaplome . . . 124
4.4 Percentage identity matrix for the five HIV-1 strains contained in the *in vitro* benchmark mix . . . 132
4.5 The five HIV-1 genes, HXB2 co-ordinates and properties of the aligned reads . . . 133
4.6 Summary of haplotypes recovered by `Gretel`, for each targeted *HIV-1* gene and strain 135
4.7 Statistics for `MEGAHIT` assembled read data from `DESMAN` preprint . . . 138
4.8 Statistical summary measuring identity of the `Gretel` recovered haplotypes to known *E. coli* COGs . . . 141

5.1 rRNA sample manifest for aliquots used to generate cDNA and amplify `Gretel` predicted candidates . . . 145
5.2 Summary of Sanger sequencing results for `Gretel` candidates . . . 151
5.3 Summary of Nanopore sequencing results, describing the number of reads that aligned to `Gretel` candidates amplified for sequencing . . . 157
5.4 Summary of unmapped reads that could be mapped to entries in an rRNA database . 158
5.5 Summary of remaining unmapped reads that could be aligned to sequences in NCBI `nr` 158
5.6 Summary of remaining unmapped reads that could be aligned to the original assembly 159
5.7 Information on the 10 genes that were selected from 259 possible candidates for *in vitro* verification of our `Hansel` and `Gretel` haplotype recovery framework. . . . 162

6.1 The 41 chosen Hungate1000 reference sequences employed for my "landscape pilot" 165
6.2 Metadata for samples selected to conduct the rumen landscape. . . . 166
6.3 Table indicating on which species we could identify `Gretel` candidates G90, G123 and G31 . . . 183

B.1 GoTaq reaction mix, used for initial exploratory PCR . . . 230

B.2 Thermocycler program for initial exploratory PCR 230
B.3 Facility template quantity guidelines for use of ABI 3730 DNA analyser 236
B.4 Facility guidelines for preparation of Sanger sequencing samples 236
B.5 Phusion reaction mix, used for high-fidelity PCR 238
B.6 Thermocycler program for high-fidelity PCR 238

List of Listings

3.1 An example `gretel` command 107
4.1 An example `shredder` command 112
4.2 An example `snpper` command 113
4.3 An example of the `NEWICK` formatted tree provided to `seq-gen` for the generation of synthetic metahaplomes 116
4.4 `bowtie2` parameters used to align generated reads to the pseudo-reference more permissively 128
A.1 An example `FASTA` file consisting of two DNA sequence records. 218
A.2 An example `FASTA` index. 219
A.3 An example `FASTQ` file containing a single read from a real sequencing experiment. . 220

Glossary

alignment the problem of identifying regions of similarity between groups of one or more biological sequences (see Section A.5).

assembly the problem of constructing large contiguous DNA sequences by exploiting the conserved overlapping regions smaller DNA fragments (see Section A.4).

BAM the compressed, binary form of the *Sequence Alignment and Map* file format (see "SAM", or Section A.3.1).

BED the *Browser Extensible Display* is a simple plaintext file format for storing and sharing coordinates for regions along a genome (see Section A.3.1).

chromatogram a line plot describing the signal observed for each of the four nucleotides, typically drawn to summarise the results of a Sanger sequencing experiment (see Section A.2.1).

dN/dS the ratio of non-synonymous to synonymous changes in a DNA sequence.

FASTA a plaintext file format typically used to store and transfer DNA and amino acids sequences (see Section A.3.1).

FASTQ a plaintext file format typically used to store and transfer sequenced reads and their associated per-base quality scores (see Section A.3.1).

GFF the *General Feature Format* is a plaintext file used to store annotations of a genome.

Hamiltonian path a path through a graph that visits every vertex only once.

Hamming distance a measure of distance between a pair of strings (see Section 4.1.2).

haplotype a set of DNA polymorphisms that occur together on a gene, or chromosome.

hb (haplotype basepair) used to refer to a basepair of one haplotype at a given position that is occupied by multiple haplotypes.

isozyme an enzyme with identical function but a different amino acid sequence to another enzyme.

Markov chain Monte Carlo an approach for efficient sampling from a probability distribution.

NP-hard a class of computational problems that are intractable, such as single individual haplotyping and *de novo* assembly (see Aside 2.1).

random access the ability to immediately access any record in a data set without the need to conduct a thorough search, typically acheived by indexing.

read a string of nucleotides emitted by a DNA sequencer.

SAM the *Sequence Alignment and Map* file format, for the storage of sequenced read alignments against a reference (See Section A.3.1).

strongly connected a graph is strongly connected if every vertex can be reached from any other.

switch error a measure of accuracy for diploid haplotyping algorithms, describing the number of times the pair of haplotypes recovered incorrectly "cross-over".

VCF the *Variant Call Format* is typically used to enumerate the locations of polymorphisms across a set of samples (see Section A.3.1).

Acronyms

aa amino acid.

ASIC application-specific integrated circuit.

bp base pair.

CCS circular consensus sequencing.

cDNA complementary deoxyribonucleic acid.

cfu colony forming units.

COGs clusters of orthologous groups.

dN rate of non-synonymous change.

dNTP deoxynucleotide triphosphate.

dS rate of synonymous change.

EC enzyme commission (number).

hb haplotype basepair.

HGT horizontal gene transfer.

LCA lowest common ancestor.

MAG metagenome-assembled genome.

MCMC Markov chain Monte Carlo.

MEC minimum error correction.

MSA multiple sequence alignment.

NOGs non-supervised clusters of orthologous groups.

nt nucleotide.

PUL polysaccharide utilization loci.

rRNA ribosomal ribonucleic acid.

SIH single individual haplotyping.

SMRT single molecule real-time sequencing (Section A.2.3).

SRA Sequence Read Archive.

SRR Sequence Read archive Run accession.

ssDNA single-stranded deoxyribonucleic acid.

VQSR viral quasispecies reconstruction.

Chapter 1

Introduction

It is hypothesized that 1.8 billion years ago the "invasion" of a suspected archaeal host by an α-proteobacterium gave rise to the mitochondrion [1, 2] that today powers all eukaryotic cells [3]. Given their multibillion-year headstart in evolutionary history, our environments remain dominated by bacterial and archaea [3], and our co-existence in this "bacterial world" has shaped our biology [4]. Evidence shows that 65% of our own genes originated in bacteria, archaea and unicellular eukaryotes [5] and thousands of genes linked to the susceptibility of genetic disease in humans were passed down to us by a eukaryotic ancestor [6].

This "intertwining" of animal and bacterial genomes is no coincidence; our evolution is strongly influenced by the microbial communities that have co-evolved with us [5], and although co-opting sequences from bacteria allows one to adapt to environmental changes and remain versatile [4], co-option is no easy feat. There are significant barriers to the success of horizontal gene transfer [7], and even successful transfers require considerable time to integrate into the regulatory systems of their new host [8], putting an effective response to stress outside of the time available to complex multicellular organisms. By comparison; the microbes that thrive on, in and around us, can divide, double, and roll the dice to adapt their population in as little as once every 20 to 30 minutes [9], offering the capability to respond quickly in the face of environmental stress [10].

Clearly, playing host to microbial communities can offer a mutual benefit. In exchange for a habitat, nutrition and the opportunity to specialise, a host can effectively extend its genome to include the functionality and plasticity offered by proximate micro-organisms [11]. These alliances have influenced the evolution of metazoans, shaping the development of most organ systems and particularly the gut [4], of humans and other animals alike [12]. Still today, we are learning how microbial communities impact our own health and wellbeing [5], the environment around us [10] and to what extent we can exploit the industrial potential of their functionality [13, 14]. Indeed, bacteria and archaea have had a significant amount of time to develop an expansive repertoire to share [3].

1.1 The Microbiome, Hologenomes and Pangenomes

In the aftermath of the Human Genome Project [15], it was hypothesised that the humble number of genes found (c. 20,000) could not explain all of the traits exhibited by a human. Indeed, we are each in an intimate symbiotic relationship with trillions of microbes estimated to outnumber our own cell counts by a factor between three and ten[1] [17]: our **microbiome**. It is hypothesised that these complex communities of microbes adapt under selective pressures to form a competitive microbiosystem responsible for performing specific activities such that the human genome does not need to evolve responsibility for them itself [18, 19].

Our entire evolutionary history, generational development to maturity, and ongoing daily function has been and continues to be influenced by the micro-organisms that have established communities within us [4]. Perturbations in our own microbiota can alter our development, or function, as well as influence our likelihood of contracting disease [5]. Indeed, some argue that given the critical importance these communities have on our health and wellbeing, our microbiome should be considered an organ system in its own right [20, 21][2].

Both infection and the medicines used to treat it, have a significant effect on the composition of our microbiota [24][3]. These communities have been implicated in obesity and diabetes [25], allergies and respiratory diseases [26], autism [27], neurodegradation [28], inflammatory bowel diseases [29] and responses to cancer treatment [30]. Thus, gaining a better understanding of our microbiota is in our best interest. From a human health perspective, our goal is to better understand the ecosystems that we host, to be capable of developing therapeutic approaches that disrupt these communities to restore lost or deficient functionality, or supplant members of the community causing harm [24]. Of course, given the abundance of bacteria and archaea in the world around us [3] and the role of these micro-organisms in the evolution of complex, structured communities, it is no surprise that such communities are not specific to humans alone [4].

A substantial area of our planet's surface is water, harbouring diverse communities of animals, plants and micro-organisms. Indeed, one of the first metagenomic shotgun sequencing projects was conducted on the Sargasso Sea [31]. The varied and often extreme conditions of the marine environment exert a selective pressure on their populations to equip themselves with mechanisms to adapt. To survive, organisms must cope with extremes in temperature, pressure, salinity, pH, light and availability of nutrients [32]. For example, seaweed grazers such as the blue-rayed limpet (*Patella pellucida*): a mollusc commonly found grazing on brown algae (typically *Laminaria digitata*) play host to microbes adapted to produce hydrolytic enzymes capable of efficiently digesting the complex carbohydrates accumulated by the seaweed [33]. *Hirondellea gigas*, a scavenger species adapted to

[1] Revised estimates indicate the numbers could be sensitive enough that a "defecation event may flip the ratio"... [16]

[2] This is not a universal opinion, with opponents arguing an organ system is composed of cells with the same genome [22]. In my opinion, if we're offering transplants of microbial matter for treatment of *C. difficile* [23], it is an organ.

[3] The role of the microbiome in the health and disease of humans has been reviewed excellently by Scotti *et al.* [24]

survive the extreme pressure found at the deepest depths of the Mariana Trench (10,994m) hosts a microbiome capable of degrading debris deposited on deep sea floors *in situ* [34].

Recently, a particularly intricate symbiosis has been characterised in coral reefs [10]. The health of structural coral (*Scleractinia*) has been observed to be highly dependent on the microbial communities that live in and around it (the coral mucus layer features a 3-4 order of magnitude more colony forming units (cfu) per ml than seawater) for its ongoing survival [35]. For example, members of the unicellular algae genus *Symbiodinium* are capable of fixing carbon via photosynthesis, transferring organic carbon to their coral host, providing a vital source of energy, and generating oxygen for respiration of the coral and its symbionts [10]. Coral bleaching, the most serious global threat to reefs (with estimates indicating up to 60% of reefs will be lost to bleaching by 2030) [36], is the result of disruption of the symbiotic interaction between hosts and the *Symbiodinium*. Owing to the vital nature of this dependence, researchers in the field have begun to consider the union between the host genome and the genomes of its associated microbial symbionts as one: the **hologenome**[4] [10, 22].

The evolutionary theory of the hologenome considers that all animals and plants establish symbiotic relationships that are inherited by future generations of the host, establishing a *holobiont* organism. These symbioses affect the fitness of the holobiont in its local environment, with variation in the hologenome (the sum total of the genomes of the symbionts) permitting rapid responses to environmental stresses or changes in the host [37]. With the fast generation time and plasticity of prokaryotes, the combined holobiont has greater adaptive potential than the host alone [10]. Indeed, the symbiont communities of the coral have allowed it to adapt to changes in sea level, pH and temperature for the past 500,000 years [36], in a faster, more versatile fashion than waiting for natural change and selection in the host genome [10].

As a theory, it remains a debate as to whether humans and other animals and plants are all holobionts, or whether the theory is suitable for consideration at all [38]. Personally, I would argue that the resilience of the function of the mature[5] human gut after significant disruption from (for example) antibiotics [39, 40] indicates some level of detachment from specific micro-organisms, as such a disruption to the community composition does not result in death of the host. Perhaps we could consider holobionce by how dependent a host is on the stability of its associated microbiota?

Regardless of whether we consider a "hologenome", or simply a collection of microbial genomes known to be associated with a host, we can only tell part of the story, with studies describing holobionts restricted to discussion of the species level [37]. Yet, with the falling cost of DNA sequencing over the past decade [41], it has become feasible to sequence multiple, distinct strains of a species (assuming they could be isolated, of course; Section 1.3); revealing intraspecific diversity across the genomes of prokaryotes [42], that is otherwise lost by considering the species alone.

[4] From the Greek *holo-*: whole or entire

[5] The infant gut is however, somewhat more vulnerable after disruption [39, 40]

A 2005 study of six pathogenic strains of *Streptococcus agalactiae* found that the "*core genome*", consisting of gene families shared by all isolates did not fully describe the repertoire of genes available to the species as a whole [43]. Even with six newly sequenced strains, and two existing database strains, the authors could still identify genes unique to a particular isolate, and extrapolated that more would still be found with the sequencing of additional strains in future. The authors argue that to truly understand a species, multiple distinct isolates are required to build a complete picture of the genes made available to the species as a whole; whether they are "core" to the survival of all strains, or an "accessory" [44] kept to just one, or a small subset of the strains of a species. Literature has come to describe the union of functionality at the disposal of a species as its **pangenome**[6] [43].

The discovery of large amounts of intraspecies diversity amongst strains has been surprising; Tettelin *et al.* calculated that for some species, even hundreds of sequenced genomes may not be sufficient to completely characterise their pangenome [43]. It has even been suggested that if one were to consider the bacterial domain as a whole, it would yield an infinitely sized pangenome, with each new isolate increasing the size of the accessory genome [45]. Though, there does exist great variation between the proportion of genes that are core between pangenomes of prokaryotic species, with the percentage of the pangenome that is core ranging between 3% (*Escherichia coli*, n=2085) and 84% (*Chlamydia trachomatis*, n=67) [42].

At first thought, it appears that the openness of a pangenome can be intuitively explained once we consider the sheer number of previously unidentified genes found by environmental shotgun sequencing (*e.g.* the Sargasso Sea [31]), the variety of ecological niches that exist within these environments [46] and the plasticity of prokaryotes in general. However, it has been difficult for evolutionary microbiologists to reconcile the existence of widely 'open' (large accessory genome) pangenomes that seem to appear in contradiction with current models of molecular evolution. Under current models, the cost of maintaining a function and a general bias towards deleting neutral alleles should preclude the opening of a pangenome [42]. Though, it has been hypothesised that species with particularly large populations (*e.g. E. coli*) and the ability to migrate between niches can overcome some of these obstacles to support a more open pangenome [42, 47].

I like to imagine that the openness of a pangenome indicates "room to manoeuvre" for a species to quickly respond to perturbations in the currently occupied niche. Accessory genes afford evolutionary opportunity, permitting sequence changes that could be advantageous under new conditions in future.

Of course, the impact of microbial communities is not limited to the health and disease of our own bodies, or the other animals and plants that inhabit our planet. Although these micro-organisms have had billions of years to occupy themselves by exploring "almost every conceivable metabolic niche" [3], the evolution of vertebrates provided a lucrative opportunity for further, novel specialisation [4]. These communities are responsible for catalyzing an incalculable number of chemical reactions in every environment imaginable [14].

[6] From the Greek *pan-*: all, inclusive of all members in a set

1.2 The Microbiome, a source of 'interesting' enzymes

All cells use polymers of amino acids called **proteins** for various purposes including structural and chemical. Proteins fold into a 3-dimensional shape with active sites to bind to other molecules with high specificity, becoming **enzymes** – catalysts to a particular chemical reaction [48].

Coding DNA is transcribed into single-stranded **messenger RNA** (mRNA) which is parsed in triplets called **codons**. Each codon corresponds to one of 20 **amino acids**: the building blocks of proteins[7]. The "central dogma of biology" can be surmised "DNA makes RNA and RNA makes protein" [49].

As introduced earlier, the microbial communities that surround us are under constant pressure to adapt to environmental stress or changes in their host [10]. A potential consequence of this pressure is the production of novel enzymes conferring the ability to perform some chemical process that allows an organism to thrive in a particular environment, outcompete others at a particular task, or destroy its competition entirely [50]. For many bacterial species, their short generation time and large population sizes introduce many unique mutations that can overcome genetic drift [42] and continue to be maintained by negative frequency-dependent selection[8] [50]. The pangenome of a species demonstrates that strains within these communities are capable of diverging quite drastically from one another, stockpiling accessory genes and enzymes that may improve their fitness [42].

The expansive repertoire of naturally occurring enzymes provides us with a superstore of versatile catalysts capable of performing a wide range of reactions on complex substrates, that have application in all facets of modern industry [14]. Exploiting these natural catalysts in industry is not a novel idea; we have been employing the use of 'extracts' for thousands of years for the preservation of meat and milk, and to produce fermented goods such as bread, cheese and beer [51]. Although these extracts were named '*enzymes*' in 1877 [52], the biomechanical process by which they worked remained mostly a mystery until the 20th century, during which study intensified and earned multiple Nobel Prizes in Chemistry including; in 1907 for cell-free fermentation [53], 1946 for crystallization of an enzyme [52] and 1984 for the first synthesis of a protein [54, 55]. Throughout the 20th century, enzymes were leveraged for large-scale fermentations of industrial solvents, acids and medicines [51].

One of the first and arguably most successful industrial enzymes isolated from nature was a thermostable DNA polymerase from *Thermus aquaticus* [56], revolutionising molecular biology by permitting automation of PCR [57]. Ever since, we have been on the hunt to take nature's best enzymes for exploitation in industrial applications and medicine [14], which are now involved in the manufacture of hundreds of different products [58].

[7] As the set of possible codons is 64 (4 bases in 3 positions: 4^3), thus larger than the space of amino acids, multiple codons may translate to the same amino acid [48].

[8] Variation under *negative frequency-dependent selection* confers an advantage to an individual, but only while the variant is rare in the population.

Over the past fifty years, the use of enzymes has become a critical part of many segments of modern industry; including the brewing, fermenting and preservation of food and drink [59–61], improvement of animal feed utilization [62], detergent agents [63], the bleaching and finishing of textiles [64–66], improving the quality of paper pulp [67], moisturizing and whitening agents for cosmetics [68], a source of energy as biofuels [69], and mass-production of medicines [70, 71].

Enzymes are big business; the sale of enzymes for food processing, and other technical applications are both multibillion dollar industries, that continue to grow [72]. In particular, **proteases** (or peptidases) constitute over 60% of the global sales of industrial enzymes [32], and are responsible for dismantling other proteins by breaking their peptide bonds with the addition of water.

Proteases have wide application [73]; interacting with gluten to improve the quality of bread [74], controlling flavour and reducing allergens in meat and milk [75], tenderize muscle and preserve flavour in the preparation of fish and cured meats [76], contribute to the finish of wool and leather materials [77], regulate the quality of pulping [78], contact lens solutions [79], wound sterilisation [73] and the breakdown of strains in laundry detergents [80].

As an example, according to enzyme nomenclature [81] – whereby enzymes are organised into classes based on their function, and assigned a corresponding *Enzyme Commission* (**EC**) number – the proteases (EC 3.4) are part of a larger enzyme class, the **hydrolases** (EC 3).

Hydrolases break bonds between compounds with the use of water[9]. Given that water is the most abundant substance of the cell, hydrolases are involved in a considerable number of vital chemical reactions in this aqueous environment [48], and as we've seen; are also catalysts for a wide variety of 'exciting' chemical reactions across many industries. The mention of hydrolases now allows me to segue toward my PhD at hand; where I was recruited to investigate the hydrolases contained in the stomach of **ruminants**.

Monogastric organisms (such as humans) feature stomachs with a single-chamber and a relatively simple linear digestive process, versus ruminants (including cows, sheep, goats and giraffes) that instead house a four-chambered stomach with a more complex digestion process which includes the need to partially regurgitate and re-chew (Latin: *ruminare*) ingested plant matter [82]. The first and largest of the chambers in the digestive system of a ruminant, is the **rumen**: a complex and diverse ecosystem inhabited by bacteria, archaea, fungi and ciliated protozoa [82]. The rumen is a densely populated environment, a single millilitre of rumen fluid yields $10^9 - 10^{11}$ bacteria and $10^5 - 10^8$ methanogenic archaea alone (as well as flagellates, cilliates and bacteriophages) [83].

[9] From the Greek, *-hydro*: water; and *-lysis*: loosening or parting

Unlike humans and other monogastrins, ruminants are capable of sequestering nutrients from plant matter with the help of symbiotic micro-organisms occupying the rumen [82]. These resident micro-organisms have co-evolved with their host, adapting to efficiently break down ingested biomass for their host's continuing survival [4, 82, 84]. These micro-organisms have developed, and continue to maintain an artillery of enzymes capable of breaking apart the complex chains of cellulose, hemicellulose and lignin found in the cell walls of plant matter, that are more easily absorbed by the host for energy [84]. This is impressive; the plant cell wall has evolved to favour robust stability, and its complex long chains of natural sugars are difficult to break down [85]. Indeed, decomposing lignocellulose efficiently still remains a challenge for our own biorefineries [85]. If one could identify and isolate combinations of enzymes like those found in the rumen, they could be leveraged to improve the efficiency of biofuels [86].

Communities capable of producing "exciting enzymes"[10] are not merely limited to humans and other animals, the environments of soil [87, 88]; ocean water [31, 32], acid mine drainage [89] and even New York City subway stations [90] to name a few, all feature microbial communities under evolutionary pressure to innovate and tune the enzymes behind these 'interesting' chemical reactions. Currently, there is great demand to further explore these environments, to 'mine' for new enzymes that would allow modern industrial processes to reduce their cost and waste; to replace raw chemical processes that have harmful by-products or high volumes of waste with enzymatic activity allowing sustainable and 'greener' production [14]. So far, a census of the bacteria and archaea in our public databases has demonstrated that we have explored the sequences of only a fraction of the diversity that pervades the microbial worlds around and within us [91, 92], seemingly leaving a wealth of enzymes with industrial potential out of our reach. Indeed, our difficulty in collecting and examining microbial diversity has a parallel history of its own, but this has not stopped our pursuit of the unculturable micro-organisms [93], as I will now describe in my next section.

[10] Where I have arbitrarily defined "exciting" as those performing chemical reactions of particular interest to us. Note that there are also many enzymes that are not necessarily involved in adaption to the environment, and do not leave the cell.

1.3 Metagenomics, as an insight into microbial communities

Through the 19th and 20th centuries, the wealth of results from “pure culture” studies had led microbiologists to believe “that the microbial world had been conquered" [93]. But in 1985, Staley and Konopka’s measurement of activity in aquatic habitats encountered an anomaly: a four-to-six order of magnitude difference between population size estimates between plating and microscopy methods [94], now known as the “great plate count anomaly”. Suddenly the microbial world was much larger and more complex than ever realised and couldn’t just be put on a plate.

As we have seen, the microbial communities that we wish to characterise are interconnected systems of symbiosis, with high dependence preventing simple isolation and culturing of individual species, and highly variable pangenomes demonstrating that sequencing of single isolates are not necessarily representative of their species. Thus traditional single-species genomics options are unsuitable.

A little over a decade later after the anomaly, Handelsman *et al.* coined the term **metagenomics** [87]. Perhaps now better characterised as *environmental genomics*, metagenomics was defined as the study of the genetic sequences of *all* micro-organisms present in an environment; such as soil, seawater or skin, rather than just one being. Handelsman *et al.* described the emergence of a “new frontier of science that unites biology and chemistry”[11], that could explore the unknown genomes of entire microbial communities found in soil, without culturing the microflora first. Although this was not the first example of environmental genomics *per se*, the term itself appears to mark the field’s shift in understanding that was 25 years in the making [93]: an acknowledgement that these micro-organisms co-exist in functional communities that cannot be readily grown in pure culture, and that it is both useful and practical to instead consider them in whole, rather than alone as individual species [95]. Torsvik and Øvreås would later echo results from the 1990s and state that fewer than 1% of soil microbes that are visible under a microscope can be characterised by culture [96], reiterating a need for approaches that could avoid difficult culturing.

Handelsman *et al.* applied cloning and screening techniques to conduct “culture-free” genomics (metagenomics), inserting DNA isolated from an environment into *Escherichia coli* plasmids, and randomly sequencing from the plasmid pool [87]. With the introduction of second generation sequencing technologies in the mid-to-late 2000s (454 and later, Illumina) enabling **whole-genome metagenomic shotgun sequencing** (Section A.2.2), metagenomic libraries could be sequenced directly without cloning [97]. Further recent innovations (Sections A.2.2 − A.2.4) have driven down the cost per megabase to mere cents and have allowed us to rapidly head toward the long-fabled “thousand dollar genome” [98].

With the increasing scale and decreasing cost of these high-yield techniques, a door to understanding unculturable microbial worlds has opened. The sampling depth and high-throughput offered by

[11] Though there was no mention of computing at the time.

modern platforms means it has become feasible to begin to sequence complex microbial populations without culture, with the limitation soon to be dependent only on the availability of both sufficient computational capacity and skilled bioinformaticians to process the output.

Genomic research is slowly progressing beyond the use of consensus DNA sequences to represent species [99], towards the ultimate goal of complete characterisation of the genetic diversity that exists across their populations. So far, research has focused on characterising specific aspects of this diversity, for example: identifying the collection of genomes across a population of micro-organisms and their host (hologenome); the gene families across all strains of a species (pangenome [100]); the groups of genes (or genetic variants within) that are inherited together in organisms across entire populations of a single species (**haplome** [101]); identifying viral strains related by mutations in a highly mutagenic environment (**quasispecies** [102]), or the sum total of genomes within an environment (the **metagenome**). However, **these modes of thinking fail to capture the true population-level variation that occurs within a gene across an entire mixed population**.

1.4 Beyond the metagenome

As we have seen, analysis of the pangenome has demonstrated that for many species, individual isolates are not enough to describe the diversity available to an entire species [43, 42]. Clearly, a reference species' genome cannot adequately represent all of its associated strains, and so neither can fragments of DNA sequenced from an environment, that have been assembled to reconstruct genomes (Section A.4) represent the full diversity within a metagenome. But microbial communities maintain a fine balance between stability and plasticity, that is driven both by their genetic breadth and diversity [103–106], and capturing all of this variation is important for our understanding of evolution and the biomining of industrially relevant enzymes [107].

The microbial communities that we share our world and bodies with are replete with diverse functions that are widely shared between bacterial strains and species via horizontal gene transfer, to the extent that it is now believed that the "tree of life" is more of a network, than a tree [108]. This intrinsic diversity and sharing leads to different species possessing genes that encode for enzymes that are capable of catalyzing the same reaction, but with the freedom to alter and fine-tune them to work under different environmental conditions, independently from other species, strains or even specific individuals within a community [14, 51]. The pangenome provides a framework to consider the full repertoire of genes available to a species, and although the literature has considered wider pangenomes, such as one encompassing entire domains of life [45], no definition has yet been proposed to describe the specific sequences of variants – the **haplotypes** – of genes shared by a community [107].

Indeed, evidence appears to indicate that possessing a gene may not be enough to confer a competitive advantage in complex microbiomes, but rather it is the breadth of functional variants of the gene at the population-level that are likely to confer advantages to individuals who have them [42, 109, 105]. This indicates the existence of important population variation at the gene-level, that cannot be represented by the pangenome alone. The existence of microbial species with small core genomes, along with the recent construction of a minimal bacterial genome requiring only 473 genes [110] would imply that a majority of genes are not strongly associated to a particular group of organisms [42], further necessitating a need to describe this variation within an alternative '-ome'.

To address this, my thesis introduces a conceptual model that allows for simultaneous characterisation of all haplotypes that pertain to a specific gene, from all organisms in a community. I call this, the **metahaplome**. The metahaplome provides a framework to describe the diversity of a particular function within a community, regardless of species or strain. I will later present the concept with a formal mathematical definition to support this change of thinking in Section 3.1. To illustrate my point briefly and place this thesis into context, Figure 1.1 depicts a "tree (or network) of life" demonstrating where the metahaplome fits in with the concepts of the hologenome, pangenome and metagenome. Imagine each of the four coloured lines as a distinct gene, perhaps encoding for enzymes that perform catalytic reactions of interest in a microbiome. The lines depict the evolutionary history of the four genes, showing their descent across species A, B and C, as well as horizontal gene transfer events (horizontal lines between tree branches). The hypothetical species are composed of individual organisms from multiple strains (not shown), who have inherited or acquired a DNA sequence for one or more of the four genes; whose combination of variants describe an **enzyme haplotype**. The metahaplomes at the bottom of the figure collect the haplotypes for each gene, across all the species and their strains. It should be noted that this is not merely just a case of finding orthologous genes: the metahaplome offers a more granular insight into the genetic variation of a community, collecting the individual haplotypes that exist within a population.

As well as formally defining the metahaplome, this thesis will go on to introduce `Hansel`: a data structure designed specifically for the storage of pairwise variants observed over reads from a whole-genome metagenomic shotgun sequencing experiment, and `Gretel`: a Bayesian framework that leverages Hansel to recover and rank haplotypes from a particular metahaplome.

I will evaluate my approach on simulated and real sets of genes, and show that `Gretel` can recover haplotypes with high identity, and consistently award those haplotypes the best likelihoods. I will show that `Gretel` can scale to process a real rumen metagenomic data set, and recover novel isoforms of enzymes relevant to industry. To the best of my knowledge, for the first time, this thesis will demonstrate that it is possible for the most likely haplotypes to be recovered with high fidelity from complex metagenomic samples enabling the characterisation of the true population-level diversity in microbial communities.

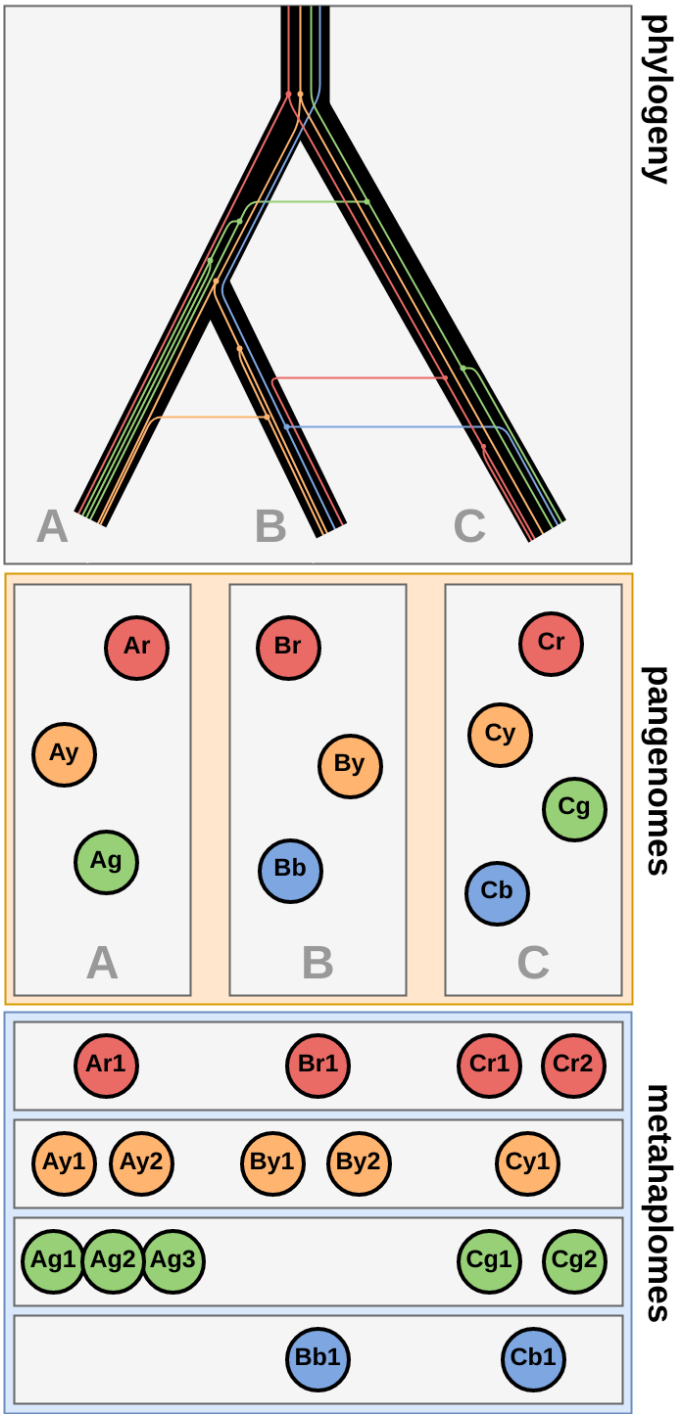


Figure 1.1 A diagram contextualising the metahaplome. The top panel depicts a phylogenetic tree showing the speciation of three species (A, B and C) composed of multiple strains (not drawn). Coloured lines represent the evolutionary history of four genes (red, yellow, green, blue), and their speciation between strains over time, and horizontal gene transfer across species from the point of view of the genes themselves (horizontal lines). The middle panel shows that the full repertoire of genes for the three distinct species can be described by their pangenomes, but fails to capture gene-level variation across the population. Finally, the bottom panel shows the **metahaplomes** of the four genes; collecting the haplotypes of each gene from the individuals in the community.

1.5 Thesis Overview

Detailed further in Appendix A, bioinformatics workflows for handling metagenomic data currently serve to investigate the distribution of taxonomy within a microbiome, and to some extent, characterisation of its function [111]. Many studies that attempt to investigate taxonomy target their analysis solely on the variation of the small subunit (SSU) ribosomal RNA gene (16S rRNA) within a sample [112]. Some, including myself, would argue that these studies would be better described as **metataxonomics** studies, given that they do not employ sequencing of the metagenome itself at all [113]. Semantics aside, these studies can only identify species (and sometimes strains) whose rRNA gene appears in a high-quality reference database [114] (such as `SILVA` [115]), which also biases the selection of appropriate primers [116]. Additionally, as the variation over each of the nine hypervariable regions (V1-V9) of the 16S rRNA gene are capable of segregating different bacterial species [117], a study incurs bias by its selection of which hypervariable region(s) to target.

Such metataxonomic studies also encompass those that attempt to broadly classify taxonomy by aligning actual shotgun metagenomic reads to databases of known sequences [112]. This can range from classification with new online databases such as `Taxonomer`[12] [118], 'ultrafast' k-mer based *lowest common ancestor* (LCA[13]) classification with `kraken` [120], or the use of long-established `blasts` against NCBI's non-redundant nucleotide database, or `RefSeq`. More recently, methods that construct phylogenies of known reference sequences to produce sets of markers capable of discerning clades in that phylogeny [121], have come into common use (Section 2.8.1).

The marker-based methodology has gained particular traction with metagenomic studies, as they permit one to build custom databases of markers specific to species or strains of interest and rapidly profile their presence and abundance in a sequenced community; many of the approaches designed specifically for metagenomic data that I will survey (Section 2.8) employ some form of marker-specific analysis. Like the contemporary method of rRNA identification just described, marker-based methods are clearly limited by the availability of high-quality reference sequences for species and strains of interest. Of course, even if we can identify a group of markers that can discern between targeted organisms, their use restricts our biological understanding of a community to discussion of only those markers alone. I argue that these 'marker-level' analyses yield insufficient resolution to provide true understanding of the population-level variation that drives the function of a community.

Simple alignment-based homology searches also drive the methodology behind functional assessment of a metagenome [112]. Even large-scale landmark projects typically rely on software designed for single-organism annotation (with or without minor modification) [122]. As I will describe throughout

[12] `Taxonomer` is available via http://taxonomer.iobio.io, constructed by the Marth Lab (http://marthlab.org/), who also produced other useful and well-designed online tools including `bam.iobio` and `gene.iobio`.

[13] Consider a pair of nodes u and v, in a tree T. The lowest common ancestor (LCA) of u and v is the deepest (farthest from the root) node of T, that is a parent of both u and v. *i.e.* If one were to retrieve two taxonomic predictions u and v, the LCA identifies the most specific taxon for which they are both descendents. This allows post-processing assignments of reads or entire contigs to increase confidence and filter spurious assignments [119]

my review, one's choice of tools can confound our insight on the real variation in a metagenome by discarding data that contradicts the assumptions of a single (often diploid) input genome.

Examples of tools available for taxonomic and functional analysis of a metagenome include the "Metagenomics Toolkit" (`MGKit`) [123], which amongst other features, can orchestrate parallel `blast` searches, or prediction of genes from custom `HMMER` profiles [124] and automatically enrich the annotations with public databases; `HUMAnN` (The HMP Unified Metabolic Analysis Network) [125]: a pipeline built as part of the Human Microbiome Project (HMP) for determining the abundance of gene families involved in metabolic pathways of interest, for a given microbial sample; or `MG-RAST` [126], a free online service for metagenome annotation built on the `SEED`[14] framework [128]. Even these functional annotation systems are limited by the availability of references. `MGKit`, `HUMAnN` and `MG-RAST` have external dependencies on public databases and the sequences within.

In general, our databases are biased toward model organisms, easily isolated bacteria and particularly well studied microbiomes such as the human gut [116, 129–131]. Multiple studies have reported that large proportions of raw reads, and components of *de novo* assemblies derived from them cannot be annotated by current workflows [112]. Other studies have shown that highly conserved functions such as housekeeping genes are often over-represented in annotated metagenomic data [132]. Clearly, the study of microbial communities in the microcosms in and around us have uncovered a "blind spot" in our public nucleotide and protein databases, plaguing downstream analyses with bias toward the taxa and proteins that we have discovered so far. It is this ongoing problem in metagenomics that led to the coining of the somewhat controversial phrase "microbial dark matter" [133, 134] to describe the organisms and functions missing from our analyses.

For the few well studied environments – such as the human gut – it is becoming increasingly plausible to side-step the use of assembly entirely [112], and directly use available reference sequences from organisms that have been isolated for annotation instead, as done by the recent landmark human gut landscape study [135]. Indeed, this too is slowly becoming a viable option for further study of the rumen with the recent publishing of the Hungate project's genome collection [136].

But regardless of whether a metagenomic assembly is constructed *de novo* from the raw reads, or if the reads are immediately aligned "assembly-free" to a known reference, one still relies on some form of a consensus sequence. Use of a high-quality linear reference from an isolated culture, a metagenomic assembly (Section A.4), or an assembly built from binned reads (*metagenome-assembled genomes*, **MAG**s [137]) still hamper our insight on the sequence variation between closely related individuals of a community that cannot be represented adequately by a single consensus sequence [138, 99].

[14] "The `SEED`" was first described in 2004 as a peer-to-peer environment for sharing annotations before they were ready for archiving in a national database. The `SEED` is branded as a 'subsystems' approach, where the primary unit of interest is not the genome, but sets of genes involved in some function. Confusingly, the name does not stand for anything, and was taken from a science-fiction novel [127].

There has been limited work on profiling beyond the species level of a metagenome, with literature focusing on the use of specific marker sequences (Section 2.9). A state-of-the-art method surveyed in Section 2.8.4 describes the tracking of strains across multiple samples with a set of universal marker genes. However, one should take care to note that the identification and subsequent tracking of the presence and abundance of strains, is not the same as recovering and analysing the DNA sequences that encode function from those strains. Such methods leverage universal marker sequences as a 'barcode' to determine the presence and quantify the abundance of species and strains in a metagenomic sample, but cannot return haplotypes.

1.5.1 Terminology

It is important to clarify the meaning of some core terms that will be used throughout this thesis. 'Haplotype' is an overloaded technical term [139], which I choose to generalise as the combination of nucleotide variants that co-occur on a molecule of DNA. As I will describe in my review, the problem of single individual haplotyping (SIH, Section 2.2) concerns itself with determining the pair of haplotypes for a human – the two DNA sequences which correspond to the halves of each chromosome pair – sometimes referred to in the literature as **global haplotyping** [140]. In contrast, **local haplotyping** refers to the recovery or reconstruction of part of the DNA sequence for a molecule. As I will discuss in my review, local haplotyping methods are typically employed as a means to reduce the complexity of global haplotyping – breaking the molecule's sequence into "local" windows, with a view to join the local haplotypes together to reconstruct the global haplotypes.

Of course, haplotyping is not limited to diploid organisms, and can also refer to work that attempts to recover the DNA sequences that make up the genomes of polyploid organisms (Section 2.6). Related to the problem of single individual haplotyping, is viral quasispecies reconstruction (VQSR, Section 2.7). VQSR considers the problem of recovering the DNA sequence 'haplotypes' for each of the highly related mutagenic strains in a viral community. Recent work describes the resolution of strains from complex microbial communities (Section 2.8), but these limited analyses refer to the distinct instances of a few short "local" marker genes as haplotypes.

These alternative interpretations of 'haplotype' seemingly exist in conflict with one another. In this thesis, I present a framework and method for the recovery of the set of DNA sequences that correspond to isoforms of a gene – **enzyme haplotypes** – on the genomes of individuals in a microbial community: the **metahaplome**. I will broadly consider a **metagenome** to be the complete set of whole genomes that compose a microbial community, which cannot be adequately inspected by isolation and culturing alone. The mathematical definitions I will present in Section 3.1 allows one to generalise the problem of recovering haplotypes using reads sequenced from a microbiome. These reads are the only evidence of the true underlying metagenome of the community. My proposal for the **metahaplome** provides a framework that unifies the problem of recovering enzyme haplotypes from viral collections, clonal bacteria populations and complex microbial communities. This thesis

will demonstrate application of my method to recover haplotypes of enzymes from synthetic gene-haplotype sets, real sequenced reads from a HIV-1 virus mix, mock microbial communities and finally, bovine and ovine metatranscriptomes.

One may argue that the targeted recovery of enzyme haplotypes is limited, as a form of local haplotyping. In this thesis, I will propose that recovery of the metahaplome is not only a novel, biologically relevant problem that requires solving, but one that allows us to sidestep the complexity of global haplotyping and simultaneously reduce our reliance on the correctness of an assembly or reference. To be clear, this work does not attempt to solve the global haplotyping problem formulated by SIH or VQSR, but to instead find the individual haplotypes of a region of interest in a microbiome's underlying metagenome. Towards the end of this thesis, my "Landscape Pilot" study will demonstrate the insight into the genetic variation of a microbial community that can now be obtained by the availability of haplotypes.

Our field's current focus is to identify ***who*** *is in the microbiome?* and summarise ***what*** *are they doing?* However, studies are restricted to analysis of distinctive 16S (small subunit) ribosomal RNA genes in a sample [111], or limited numbers of marker genes. We are not yet able to routinely analyse the true genomic variation across individuals of a community [111, 141]. Broadly speaking, I like to think that the work presented in this thesis will bridge this gap, and start asking ***how*** *are they doing it?* **What variants (*isoforms*) of genes exist in these communities that allow them to perform interesting tasks,** such as the breakdown of biomass in the rumen? I *choose* to study "local" haplotypes, and will present my arguments that for a metagenome, these enzyme haplotypes are more useful and informative, than the recovery of entire strains.

In the following Chapter, I will turn my attention specifically to the history and state-of-the-art of algorithms designed for haplotype recovery, since the problem of reconstructing the two haplotypes of a diploid human was first formally defined in 2001 [142]. I will show the difficulties encountered in the literature as recent focus shifted away from humans (diploids), to polyploids (Section 2.6), viruses (Section 2.7) and metagenomes (Section 2.8), and that until now, no algorithm has been capable of recovering haplotypes from a metagenome.

1.5.2 Contributions

A Review of Haplotype Recovery

I provide an in-depth review of algorithms introduced by the haplotyping literature, covering the history of haplotype recovery since the problem was first introduced in 2001, comparing and contrasting the different ideas and implementations, and demonstrate why current approaches have been unable to consider haplotype-level variation in metagenomic data sets.

The Metahaplome

This work introduces the metahaplome as the set of haplotypes for a particular region of interest (such as a gene that encodes for an enzyme) within a metagenome. As a conceptual model for the application of haplotyping to a microbial community, I will demonstrate how it can be used to explore the variation observed for a particular gene, throughout a population.

Hansel and Gretel: a framework for recovering haplotypes from a metagenome

This work outlines `Hansel`, a data structure designed to efficiently store variation observed across sequenced reads, and `Gretel`, an algorithm that leverages `Hansel` for the recovery of haplotypes from a metagenome. My method:

- recovers haplotypes from metagenomic data
- does not need *a priori* knowledge of the number of haplotypes
- makes no assumptions about the distribution of alleles at any variant site
- does not need to distinguish between sequence error and variation
- uses all available evidence provided by the raw reads
- does not require any user-defined parameters
- does not require bootstrapping, model building or pre-processing
- can confidently rank its own results based on calculated likelihoods
- can be executed on an ordinary computer

I will show *in silico* that `Gretel` is capable of recovering haplotypes from short reads with high similarity to the correct haplotypes, and consistently award the best recovered haplotypes with the highest likelihoods. I will demonstrate *in vitro* verification of my method, showing that the haplotypes recovered by `Gretel` from short-read Illumina sequencing of a rumen metatranscriptome, are well supported by Sanger sequencing, and have high affinity to reads generated via Nanopore sequencing: contributing the **first computational recovery of novel enzyme isoforms** from a real microbiome. Finally, I introduce **the first pilot haplotype-landscape study** of a real microbiome.

1.5.3 Chapter Descriptions

The goal of this thesis is to provide a formalisation, computational algorithm and verification thereof, for a method to recover haplotypes from a sequenced microbial community. As a joint computer science and biological sciences PhD, my work may be arranged in a fashion that differs from what the reader may be accustomed to, and is divided into six further chapters:

- **A Review of Haplotype Assembly**
  The *Introduction and Background* has so far given a brief contextual overview for this thesis, setting the scene of these microbial communities and introducing current limitations for investigating and discussing the DNA sequences within. Chapter 2 will embark upon a detailed, algorithmic-level review of computational haplotyping, since it was first proposed in 2001. Additionally, Appendix A may serve as a glossary and help to provide some additional context for this thesis, particularly for those who are not familiar with bioinformatics.

- **Theoretical Work**
  Chapter 3 will introduce the metahaplome (Section 3.1), providing my justification for why one should use it to consider the variation across genes within a community without needing to first consider taxonomy. Sections 3.2 and 3.3 will provide the intuition, algorithms and equations for `Hansel` and `Gretel`.

- ***In silico* Evaluation**
  Chapter 4 will cover the *in silico* testing of the `Hansel` and `Gretel` framework. I will briefly introduce two utility programs I wrote for generating simple short reads, and variant calling (`shredder` and `snpper`, respectively). I will then cover evaluation of my method on synthetic reads generated by simulated evolution, from a set of real *DHFR* genes, a real HIV1 sequencing sample, and a synthetic microbial community.

- ***In vitro* Evaluation**
  Chapter 5 describes my *in vitro* evaluation of `Gretel`; introducing the rumen metatranscriptomic data set from which haplotypes were computationally recovered. I will describe my wet laboratory work, where I performed gene-specific reverse transcription and PCR for selected candidates, whose sequences were validated with Sanger sequencing, and verified with nanopore strand sequencing.

- **The Rumen Landscape Pilot**
  Having demonstrated the validity of the approach, Chapter 6 introduces a pilot data set as a means to show the utility of `Hansel` and `Gretel` for generating novel biological insight.

- **Discussion**
  Finally, I close my thesis in Chapter 7 with a discussion of the efficiency of the method, a comparison to methods previously reviewed in Chapter 2, and the future direction of my work.

Chapter 2

A Review of Haplotype Assembly

At the start of the millennium, the completion of the first human whole-genome sequence confirmed that our genomes are well-conserved. Small regions of our DNA are responsible for much of the diversity observed within our species. The smallest unit of this variation is the single nucleotide polymorphism (SNP), at which the nucleotide base in a diploid can be one of two alleles.

The human genome is organised in pairs of chromosomes: we are *diploid* organisms. Although our genes and their loci are homologously paired, a single individual can be *homozygous* or *heterozygous* at a given position of the genome. That is, at a SNP, nucleotides at a specific locus on the two chromosome pairs may be different from each other. Thus, a single individual's genome can be described as two sequences of SNPs, demarcated by chromosomes. These sequences may be defined as *haplotypes* (haploid genotypes).

SNPs are fundamentally important across the tree of life, and can shed light on how a species has evolved, and offer a means of disease diagnosis and individualised medicine. In particular, one may wish to know which specific variants along some region of the genome were inherited together.

As chromosomes tightly link genes and their associated alleles together, and operate as a vehicle for inheritance [143], one's haplotypes can describe how alleles were inherited. From a human-centric perspective, *haplotyping* refers to the process of deriving the two haplotypes for a given diploid genome [144]. But for a metagenome, the term *haplotype* is more difficult to define. It has been shown that despite variation in community composition in extensive sampling of gut microbiota between individuals, function is remarkably well conserved [145]. This single finding will later underpin my argument in Section 3.1, that to truly characterise the diversity of a microbial community, it is necessary to look beyond taxonomy and the genomes within, and instead look at the collection of genes – and more importantly – their variants that are shared by a community. But first, through this Chapter I will describe the historical motivations behind computational haplotyping, and its influence on modern methods for virus and strain recovery[1].

[1] The impatient reader, may wish to jump to Section 2.8, where I focus my discussion on complex microbial communities.

2.1 Introduction

The computational problems involved with haplotyping arise as DNA preparation and sequencing technologies have not yet reached the capability of accurately sequencing entire genomes with high coverage. This is especially problematic for metagenomics, where deep coverage is essential to observe sufficient sequence from as many of the constituent genomes as possible [93]. Our insight to a sample, whether from a single organism, or a metagenome, is typically[2] through millions of small sequenced fragments (reads) that must be pieced together by the computational process of **assembly**.

As described in Section A.4, humans are remarkably well studied organisms. The availability of a gold standard reference sequence eases the problem of assembly, as it reduces the problem to one of correctly aligning sequenced DNA fragments against the correct position in the human genome reference. For haplotyping, via HapMap [147] and the Haplotype Reference Consortium [148], human haplotype variation is well mapped, and a single individual's pair of haplotypes can be reconstructed with sufficiently deep whole-genome sequencing, or even imputed ("*phased*" [149]) from large scale SNP panels such as those offered commercially by 23andMe. Of course, even for a genome as well studied as our own, there are still gaps in the assembly, and it should be noted I do not wish to diminish the difficulty of remaining problems, such as the major histocompatability complex, and untangling large regions of short-tandem repeats found in chromosome centromeres [146].

However, by comparison, metagenomes are largely unknown environments. Isolation of individual organisms is difficult, and the identity and even number of species in an environmental sample is often an unknown. Although reference projects such as the Hungate1000 exist for ruminants, there is still a lack of suitable reference sequences for many of the species in the rumen [136] and elsewhere [92]. Thus, for assembly in a metagenomic context, one must turn to reference-less *de novo* assembly.

Assembly exploits the high level of conservation observed in genomes to arrange sequenced fragments by their overlapping regions they have in common, in order to reconstruct larger contiguous DNA sequences. But as introduced in Section A.4, the general goal of assemblers is to generate a consensus sequence that best represents the organisms present [150]. Thus the objective of assembly is at odds with our desire to explore the haplotypes that encode isoforms of genes within a microbial community; the "metahaplome" (Section 1.4).

For the context of this work, it is important to note that the problem of recovering haplotypes from the metahaplome is different from that of assembly. Assemblers produce consensus sequences, which cannot (and do not aim) to resolve haplotypes. Even specialised metagenomic assemblers do not aim to reconstruct strain-level variation from a microbial community [151]. I will later show that my proposed method can correctly recover fine-scale population-level variation, which is simply not possible with assembly, or DNA sequencing alone [152].

[2] Though there has now been a human whole-genome assembly completed with long-read Nanopore sequencing [146]

2.2 Origins of the problem

The problem of **single individual haplotyping** (**SIH**) was first described by Lancia *et al.* at Celera Genomics in 2001 [142]. In the wake of the announcement of Celera's first human genome, it became clear that the next big research problem was not only to analyse the millions of single point variants that populate our genomes, but how to assemble the two haplotypes that make up a single individual's genome. The 2001 work introduced the first terminology and notation for "computational SNPology"[3]. Although a full discussion of the history of human (and by extension, diploid) haplotyping is outside the scope of this thesis, it is important to introduce the first description of the problem, as it forms the foundation for many other approaches and algorithms that followed, for diploids and polyploids alike.

The work first informally described the problem *for an individual*:

> "Given a set of fragments obtained by DNA sequencing from the two copies of a chromosome, reconstruct two haplotypes that would be compatible with all the fragments observed." — Lancia *et al.* (2001)

Even in an ideal scenario with error-free read fragments, lack of coverage across the haplotypes (be it a chromosome, or region of interest) in question, will obfuscate recovery efforts. So this new definition relaxes the requirement to reconstruct sequences that are "*compatible*" with observed fragments, where it is sequencing error that obstructs haplotype recovery. Thus the proposed implementations first reformulate the problem as removing the errors that *must* exist within the observed data.

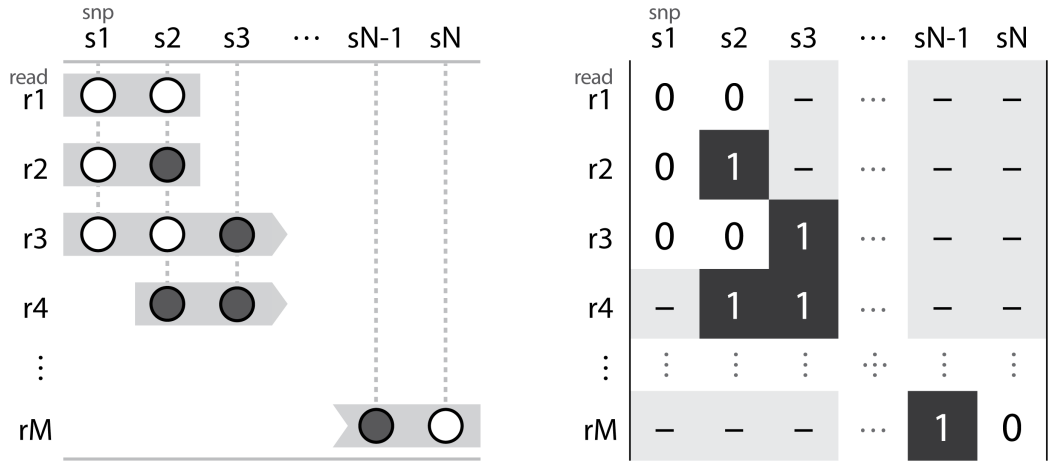


Figure 2.1 An example SNP matrix. Read fragments represented by grey boxes (left) are aligned to some reference with known SNP loci. The alleles at the SNP loci are represented by white and grey circles. These reads can be alternatively represented by an $m \times n$ SNP matrix (right). Each row of the matrix models one of the m read fragments and each column corresponds to one of the n SNPs. Elements encode the allele at a given SNP for a particular read fragment as a 0 or 1, or a $-$ if the read does not cover that position. A column containing only one element indicates the corresponding SNP site is homozygous, otherwise it is heterozygous.

[3] A phrase that did not seem to catch on in the literature

In this work Lancia *at al.* defined three optimisation problems to solve SIH:

- *Minimum fragment removal* (**MFR**);
- *Minimum SNP removal* (**MSR**); and
- *Longest haplotype reconstruction* (**LHR**).

Perhaps more importantly, in addition to the proposed optimisation problems, Lancia defined a common notation to formally describe the problem of single individual haplotyping. The **SNP matrix**[4] (typically denoted M): an $m \times n$ matrix encoding the binary allele observed at each SNP site $1..n$ on each read fragment $1..m$. That is, $M[i][j]$ is one of two possible alleles (typically labelled $0, 1$ or A, B), or a gap (denoted $-$) observed at the j'th SNP site on the i'th read fragment (Figure 2.1).

Consider a set of read fragments $R = \{r_1, ..., r_m\}$, and a set of SNPs[5] $S = \{s_1, ..., s_n\}$. Each read fragment is a row in M, and each SNP is a column. Lancia *et al.* find errors that must be mitigated by looking for "conflicts" in M. A pair of read fragments $r_i, r_j \in R$ are said to be in read or **fragment conflict** if there exists at least one SNP $s_k \in S$, such that $M[i][k]$ and $M[j][k]$ are both non-gaps, and do not agree on an allele (Figure 2.2a). That is, reads r_i and r_j have opposing alleles on at least one SNP. Similarly, a pair of SNPs $s_k, s_l \in S$ are said to be in **SNP conflict** if there exists a read fragment pair $r_u, r_v \in R$ such that the 2×2 submatrix $M[u, v][k, m]$ contains a single element that disagrees with the other three. That is, reads r_u and r_v are heterozygous at SNP s_k or s_l, and homozygous at the other (Figure 2.2b).

Although a conflict might indeed indicate that a pair of fragments originate from different haplotypes, Lancia *et al.* focus on the idea that conflicts arise due to errors in the underlying sequence data, arguing that "experiments in molecular biology are never error-free". A SNP matrix M which contains any conflicts is *infeasible*, and the three optimisation methods aim to graphically represent and resolve the conflicts in M, to yield a pair of feasible haplotypes.

Aside 2.1: NP-hard Computational Problems

Decision problems may be classed by their complexity. Briefly, problems that are computable "quickly" (in polynomial time) are classed P and problems whose solutions can be quickly verified are in NP. As solutions to P-class problems must be verifiable at least as quickly as generating them (by repeating the computation), all problems in P are in NP. However, a set of particularly difficult (**intractable**) problems that all NP problems can be reduced to are classed as **NP-hard**.
For example, the *travelling salesman* problem, to determine the shortest route that visits a set of cities once (by finding the minimum weight path through a graph of connected cities) is NP-hard. Determining the shortest path through an overlap graph for assembly, or finding the optimum elements of a matrix to remove or flip to find haplotypes are also NP-hard. It remains to be proven whether polynomial-time solutions for NP-hard problems exist (or not). Since posed in 1971, the question of whether $P = NP$ (or not) is one of the most important open problems in computer science and mathematics.

[4] Defined by Lancia *et al.* "...in the obvious way"[142]
[5] I will refer to "SNP" as a shorthand for a particular SNP site.

2.2.1 Minimum fragment (MFR) and minimum SNP (MSR) removal

We may consider the first two optimisations together. Minimum fragment, and minimum SNP removal, both focus on the identification and reconciliation of one of the two types of conflicts in a SNP matrix. Lancia *et al.* suggested that read fragments could be 'bad' due to contamination and multiple read errors, and that a SNP may be 'bad' if multiple fragments at that position contain read errors. The earlier problem definition was once again reformulated:

> Find the minimum number of fragments, or the minimum number of SNPs to ignore, so that the (corrected) data is consistent with the existence of two haplotypes measured by error–free DNA sequencing. — Lancia *et al.* (2001)

Minimum fragment removal aims to remove the minimum number of rows from M such that the matrix becomes feasible, whilst minimum SNP removal aims to remove the minimum number of columns from M to achieve the same objective. Both approaches build a graph from the contents of M and perform operations on the resultant graph as a proxy to M.

MFR builds a **fragment conflict graph** $G_F(M)$, where fragments pairs are joined by an edge if they are in conflict. Similarly, MSR constructs a **SNP conflict graph** $G_S(M)$, where SNP pairs are joined by an edge if they are in conflict. Figure 2.2 depicts a simple example where a single base miscall creates conflicts in M, and the resulting conflict graphs $G_F(M)$ and $G_S(M)$.

If the read conflict graph $G_F(M)$ can be partitioned such that the vertices (read fragments) can be divided into two independent sets, where no fragment is linked by a conflict edge to another fragment in the same set (we could also just say that $G_F(M)$ is bipartite), then each set contains a group of fragments that only conflict with fragments in the other group, both of which can be merged to obtain a pair of haplotypes h_1 and h_2 that satisfy the problem.

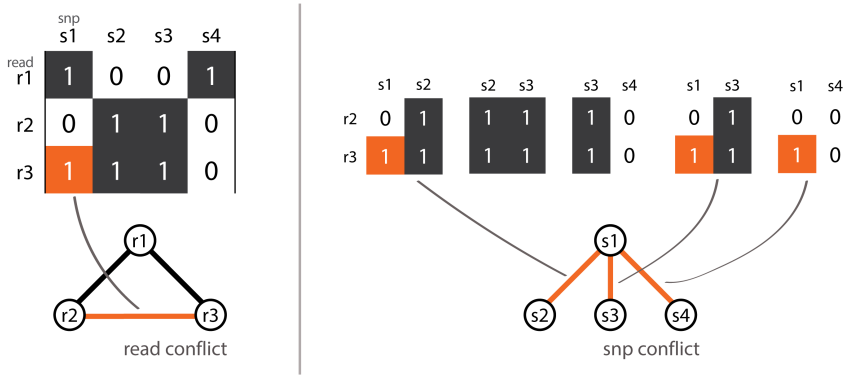


Figure 2.2 An example demonstrating how an incorrect base call at $M[3][1]$ (orange) causes a conflict under the MFR and MSR frameworks. Left, the error causes a conflict between two otherwise unconflicting reads (r_2 and r_3) in the read fragment conflict graph $G_F(M)$. Right, the error causes conflicts to arise between SNP s_1 and $\{s_2, s_3, s_4\}$ in the SNP conflict graph $G_S(M)$. Reconciling the SNP matrix M requires disposal of one or more entire reads (MFR), or all alleles at one or more SNPs (MSR), even if error is localised to a single miscall.

Additionally, if one can construct a set of vertices (SNPs) in the SNP conflict graph $G_S(M)$ such that no two SNPs in the set are linked by a conflict edge (an independent, or stable set), the article shows that the maximum independent set's equivalent representation in $G_F(M)$ must be bipartite. This latter strategy appears to indirectly solve the problem only by virtue of showing that $G_F(M)$ must be bipartite given the maximum independent set of $G_S(M)$.

Lancia *et al.* recognised that the problem becomes significantly more complex with gapped reads. Gaps may arise from sequence errors that have been masked out (*i.e.* low confidence calls), or the use of mate-pair sequencing technology. The work formally proved that in general, determining the optimum set of rows or columns to remove from M to solve MFR or MSR (respectfully) are NP-hard computational problems. Very briefly, the statement of MFR can be reduced to the known NP-hard optimisation problem of inducing the maximum bipartite subgraph, and MSR to the maximum cut (*MAXCUT*) problem [153]. Although in the case where fragments are gapless[6], both MFR and MSR can effectively be reduced to polynomial-time problems.

It is unclear why future literature favoured strategies for solving MFR over MSR, but it is perhaps owing to the more direct proof from MFR, combined with the more intuitive idea of considering read fragments in conflict as opposed to the SNPs themselves. Lancia suggested that MFR is more suited to applications where one is worried about contaminants, rather than sequencing error, and that MSR should be used in the presence of sequencing errors only [144]. However, I would argue that MSR appears to be a more aggressive method for removing observations from M, where sequenced bases from every fragment observed across a set of SNPs are discarded to construct a pair of haplotypes compatible with what is left of the evidence in M.

Further description of the proposed algorithms (including additional strategies for solving MFR and MSR later by Lippert *et al.* [153]) fall outside the scope of this review, primarily as none of them were implemented in publicly available software, and did not propagate into popular use due to the introduction of the minimum error correction optimisation less than a year later (Section 2.3).

2.2.2 Longest haplotype reconstruction (LHR)

Longest haplotype reconstruction (LHR) is an approach that imposes an extra constraint to MFR, aiming to remove rows from the SNP matrix M whilst also maximising the length of the derived haplotypes. That is, LHR aims to minimise the number of gaps in the two reconstructed haplotypes. The strategy is only briefly described in Lancia *et al.* (2001) [142] and does not feature in the follow up work by Lippert *et al.* (2002) [153]. While the problem does not appear to have been further developed as a means for solving SIH in practice, it is still one that continues to generate academic interest, including a peculiar recent proof that the problem becomes NP-hard in the ungapped case if the reads are error-free [154].

[6] "On a positive note." [142]

2.3 Minimum error correction (MEC)

In 2002, a year after the initial work by Lancia *et al.*, four of the five original authors based at Celera Genomics (including Lancia) released a follow-up article describing additional strategies for the haplotype assembly problem [153]. Whilst the work included updated descriptions for MFR and MSR (with no mention of LHR), the work also first introduced *minimum error correction*[7] (**MEC**).

Like MFR and MSR, MEC still focuses on the idea that it is sequence error that fundamentally obstructs the accurate recovery of haplotypes. However while MFR and MSR discard the evidence that support entire read fragments, or all the alleles observed over a set of SNPs for every fragment; minimum error correction is less destructive and instead aims to find the fewest elements in the SNP matrix M to "flip", such that M becomes feasible and yields a pair of haplotypes. MEC alters, rather than outright discards the observations in M, under the assumption that it is erroneous bases from sequencing error that need correcting to produce a pair of haplotypes.

Like MFR and MSR, MEC is demonstrably an NP-hard computational problem, but unlike these other optimisations, MEC is also NP-hard when the reads are ungapped [153, 156]. Despite the complexity of MEC when compared to MFR and MSR, it quickly gained popularity and a 2010 survey of SIH reconstruction algorithms identified MEC as the most commonly used approach to modelling the problem at the time [157]. Panconesi and Sozio argued in 2004 that MEC was a "more direct approach", favouring it over the "round-about [...] auxiliary combinatorial problem" of MFR [155]. I would also argue that MEC serves as a more realistic model for the correction of incorrect called or poorly sequenced bases.

MFR, MSR and MEC all pose combinatorial problems in selecting the optimum set of rows or columns to remove from, or elements to flip, in the SNP matrix M. Initial algorithms used naive and expensive branch and bound approaches to search and reduce the solution space [153]. Interestingly, a lack of available sequencing data at the time[8] made it difficult to develop, test and compare strategies for computationally generating solutions to SIH [158]. As real data became available focus quickly moved towards the use of heuristics to make the generation of approximate solutions to SIH both computationally tractable, and more accurate than the previous greedy methods [144, 157].

2.3.1 FastHare (2004)

FastHare [155] was the first heuristic for MEC, proposed by Panconesi and Sozio in 2004. The approach is simple; sort all the fragments by their starting position along the chromosome, then incrementally construct a pair of consensus haplotypes by assigning the observed fragments to the

[7] Confusingly, MEC is also sometimes referred to in the literature as *minimum letter flip* (MLF) [144] or *minimum element removal* (MER) [155]

[8] "...given the regrettable unavailability of real data in the public domain" [155]

closest of the two partial haplotypes until all fragments have been processed. The strategy also first discards from consideration any SNP site where the proportion of the minor allele is less than 0.2, the intuition being that if 80% of a column in M (excluding gaps $-$) is one allele, the opposing minor symbol is likely to be error. The columns are reinserted later, assuming the majority symbol was correct.

`FastHare`'s speed arises from this simple and greedy approach. In particular, the greedy assumption that each fragment can be coerced into one of the two haplotypes sidesteps the complexity of the previous optimisations that attempt to find the optimum set of fragments (or SNPs) to ignore. The heuristic also avoids the need to construct and manipulate a conflict graph.

`FastHare` was shown to work well with small data sets (particularly with low-error rates), **demonstrating to the research community for the first time that a heuristic was capable of providing reliable haplotype reconstructions**. The method inspired further work on approximating solutions to SIH, rather than continuing with slow and expensive branch and bound algorithms, or the use of large numbers of precomputed solutions (dynamic programming[9]). `FastHare` continues to appear in recent literature as a yardstick for baseline performance in comparison to newer methods.

2.3.2 The first diploid genome sequence of an individual human (2007)

In 2007, Levy *et al.* published a landmark work that presented the first diploid human genome reference sequence: *HuRef*[10] [159]. Briefly, the sequence was generated from 32 million sequence reads, which were assembled with a modified[11] version of the open-source Celera Assembler [160], mapped to version 36 of the NCBI Reference Assembly, and identified, filtered and validated variants. The authors identified some 4.1 million variants, 30% of which had not been previously described by other sequencing endeavours; 3.2 million of which were single nucleotide polymorphisms. Without wishing to sideline the importance and difficulty of the significant work detailed within the paper, I limit specific discussion to the novel method with which the two haplotypes were generated, for lack of a name, a later review by Geraci [157] termed Levy *et al.*'s approach **DGS**[12].

The approach involved constructing a filtered set consisting 1.85 million heterozygous SNPs for "haplotype assembly". Interestingly, fewer than 50% could be "chained" into more than 6 SNPs where

[9] Dynamic programming refers to precomputing and storing a large number of smaller solutions to a computational problem. Approaches that use dynamic programming then reduce the main problem at hand to be solvable by the subproblems whose solutions are already stored.

[10] The human was J. Craig Venter, the founder of Celera Genomics, and last author of the work.

[11] The modification is described in the Methods. Instead of a position-based column-by-column approach, the assembler was redesigned to consider heterozygous elements together, delimiting such regions in the context of 11 bp (at least) non-varying flanking regions (*i.e.* A single column SNP was considered as a single-column variant if it was flanked either side by at least 11 bp of sequence without variation across the aligned reads). Reads that supported heterozygous SNPs (or regions) were grouped into their respective alleles and the consensus sequence selected the highest supported allele given the highest quality-weighted sum of evidence across each read group.

[12] Presumably for *diploid genome sequence*, but the name is not explained.

consecutive variants were within 1 kbp of one another. On average, a pair of SNPs was separated by 1500 bp. Indeed, the authors cite a personal communication with Lippert that highlighted the expectation that observed variants would not appear in close proximity, limiting the use of short read sequencing without mate pairs in the context of humans. With the use of mate pairs, Levy *et al.* found that a single variant could be linked to 8.7 other variants on average.

For haplotype assembly with 'DGS', the data was encoded with Lancia *et al.*'s SNP matrix M, where an element $M[i][j]$; the j'th SNP on fragment i was encoded with a 0 if it matched the consensus Celera Assembler sequence, and a 1 otherwise. The algorithm considers the read fragment (row in M) that features the fewest gaps ($-$) and uses this to seed the haplotype solution h. The binary complement is then taken for the other haplotype $\bar{h}$. DGS then considers each remaining unassigned read fragment, and selects the one that maximises the difference between the number of positions it has in common between h and $\bar{h}$. The fragment with the largest such distance is then assigned to the haplotype it was more similar to; extending the haplotype. The 'other' haplotype is updated with the complementary string. When no more fragments can be assigned to either h or $\bar{h}$, the current haplotype configuration is refined by determining the majority rule of each SNP position, and the current fragment assignments are also refined given the (possibly new) haplotype assignment of SNPs along its sequence.

This process iterates until no changes are made to the haplotype configuration. Clearly the approach is greedy, and although not explicitly likened to MEC in the manuscript, one can argue that its simple strategy attempts to naively minimise the number of 'flips' that must be performed to assign the majority rule of a column or row in M. With no reference to FastHare in the work; DGS may have been the first such greedy MEC-like heuristic for reconstructing a diploid haplotype, noting that the manuscript is the culmination of ten years of work, only a small part of which concerned itself with haplotype assembly.

DGS could construct > 200 kbp haplotype blocks, spanning more than half of the *HuRef* genome. Comparisons to the HapMap data of the time showed the reconstructions to be highly accurate. Indeed, a 2010 review would later show DGS to be one of the best haplotype assembly strategies[13] [157], but (presumably due to space limitations) the Levy *et al.* manuscript does not describe the time and space complexity of the approach, and neither does the later review. One may imagine with so many pairwise distance comparisons, and the iterative refinement process, the algorithm was likely computationally prohibitive compared to some of the alternatives that appeared later. Arguably the lack of a name for the tool and its limited availability upon publication is a potential explanation for why it did not find much popularity for haplotype reconstruction in its own right, despite its accuracy.

DGS did not develop much interest in itself, but the overall work **proved that *whole-genome diploid haplotyping* was possible, interesting and had utility**; identifying many thousands of previously unseen variants, despite the availability of population studies. The work demonstrated that 44% of

[13] It is not clear how a working copy of DGS was obtained, as it did not appear to be generally available.

the annotated genes on *HuRef* have at least one 'alteration' within them; paving the way for true insight of variation on a human's genome, and offering one the earliest, greatest insights to the genotype-phenotype correlation.

Amusingly their Discussion describes the 'most significant technical challenge' was the sum of the work that led up to the assembly of haplotypes: trying to maintain the integrity of the "allelic contributions" from the reads themselves. The **quality of the sequence data, assembly and variants are credited with the accuracy given the "relatively simple means" in which the haplotypes were recovered**. Though, the work does note a future avenue for improving haplotypes that considers Markov chain Monte Carlo (MCMC) as a means to sample haplotype configurations more efficiently from the SNP matrix M – an idea realised just a year later by Bansal *et al.* in the form of `HASH` (Section 2.3.4).

2.3.3 SpeedHap (2007)

In 2005, a year after the publication of `FastHare`, the International HapMap Consortium published the human genome's "Haplotype Map" [161] highlighting a need for continued exploration of the individual haplotypes that could be recovered in sequencing data. A few years later, with publicly available real data, it became more practical to develop tools for the single individual haplotype problem. **SpeedHap** [162] was proposed as an alternative to the popular `FastHare` heuristic. `SpeedHap` was designed particularly to work in circumstances with high read error (up to 20%), or coverage was low (down to $3\times$) – which were both weakpoints for the `FastHare` algorithm.

The algorithm for `SpeedHap` can be broken into a preprocessing step, followed by three main phases. Each phase weakens the conditions necessary to update M in order to converge on a potential solution. Arguably the approach described is not attempting to optimise the minimum error correction criterion *per se*, but fits well in our narrative of the concept that M contains errors that must be flipped in order to yield a pair of haplotypes. `SpeedHap` begins by organising the non-homozygous SNP columns of the matrix M into "profiles"[14], P_i for $i = 1..n$ (recall that $|S| = n$). All P_i contain two sets of fragment indices $r_j \in R$, corresponding to the two possible alleles (referred to in the manuscript as 1 or 2), which both enumerate the fragments featuring that allele at the SNP i. For a given pair of profiles, P_i and P_k, one can define a 2×2 **conflict matrix** $E_{i,k}$. The four elements of the conflict matrix $E_{i,k}$ correspond to the four potential set intersections of the profile pair (P_i, P_k). That is, the elements contain the set of read fragments that support each of the four combinations of the two possible alleles between SNPs i and k: $(i,k) \in \{(1,1),(1,2),(2,1),(2,2)\}$.

When one diagonal of the conflict matrix is non-empty, and the other diagonal elements are $\emptyset$, it is said that SNPs i and k are conflict free. Otherwise, if there exists a single $\emptyset \in E_{i,k}$, the fragments enumerated in the set of the opposing diagonal are assumed to be in error in at least one of the two

[14] Referred to as G_i in the manuscript, but I wish to avoid confusion with previously defined graphs

SNP sites. SpeedHap constructs a conflict matrix for all column pairs in M. When there is sufficient evidence to identify both the row and column of an error in M, the element is swapped for the gap symbol ('$-$') otherwise all fragments involved in conflict have their symbols at i and k swapped to a gap in M. The profiles are recomputed for the next phase.

Following the preprocessing correction step, a graph[15] G is constructed by connecting all profiles still in conflict with an edge. That is, G is a graph where profile nodes are connected by an edge to another SNP profile if their corresponding matrix $E_{i,k}$ remained in conflict after preprocessing. The graph is reduced by removing all vertices from G whose degree is above average, effectively removing profiles involved in more conflicts than average from future consideration, yielding G_r. The complementary graph $\bar{G}_r$ features all the profiles vertices found in G_r, but now only connects them if the pair's conflict matrix is conflict free. A depth first search (DFS) is used to find the largest connected component; yielding the largest subset of profiles (or SNP columns) in M that are not in conflict with each other. The unconflicting read fragments associated with the profiles in the largest component are used to construct a pair of partial haplotypes.

The third phase selects remaining profiles that can be disambiguated given the now constructed partial solution, primarily by using the ratio of the alleles at the profiles from the previous phase to determine which remaining profiles could now be homozygous. Non homozygous column profiles that were not included in the previous step's partial solution have new conflict matrices calculated between them and the partial solution, with the goal to correct more elements of M without necessarily requiring a full rank matrix as before. The final phase completes the solution, attempting to merge any final unused information with the haplotype pair if possible. If conflicts remain, the evidence is effectively removed from M. Two consensus haplotypes can then be constructed with M.

The manuscript presents favourable results over FastHare[16], noting better results in cases with high error and low coverage, and similar results under less extreme conditions. Though in a later review of the literature [157], Geraci fails to find improvement in accuracy by using SpeedHap over FastHare.

2.3.4 HASH (2008)

A few years after Wang *et al.* had written-off the performance of Markov chains for haplotype recovery (discussed later in Section 2.5.2), Bansal *et al.* proposed *Haplotype Assembly for Single Human* (**HASH**), an MCMC-based probabilistic[17] heuristic for MEC [163]. The goal of HASH is to sample from the distribution of all possible haplotype pairs H, while maximising the likelihood of a pair of haplotypes h, given the SNP matrix M (and a corresponding matrix of per-base error probabilities if available).

[15] Referred to as C in the manuscript, contrary to other work that constructs a graph from a derivative of M

[16] Amusingly, the authors had to implement their own version of FastHare from the algorithm in the manuscript, as making software available was somewhat difficult in FastHare's time.

[17] HASH is likelihood based, but still attempts to optimise the MEC criterion, and so features here instead of in my later section on probabilistic approaches.

Consider a hypothetical strongly connected[18] graph G_H, defined such that every possible haplotype pair ($h \in H$) has a node in the graph. Edges in G_H enumerate the available transitions between a given h and all possible derivative haplotype pairs that could be generated by 'flipping' the bits of the haplotypes in h, for each available subset of the SNP columns in M. That is for a given h and a subset of SNP columns C ($\subset S$), there exists a transition edge $h \rightarrow h_C$. Note that C may be $\emptyset$, permitting an edge that joins h to itself. Succinctly, a given h can be considered as a state in a *Markov chain Monte Carlo* (**MCMC**) system that can transition to any other state (including itself) with some probability.

Naively, one could construct a set of SNP column subsets $\Psi_1 = \{\emptyset, \{1\}, \{2\}, ...\{n\}\}$ (recall $|S| = n$), such that a haplotype state h has $n+1$ possible transitions, whereby only a single column will be flipped when moving to a different state. However the authors show this method typically converges on the most probable solution at an exponentially slower rate as the sequencing depth increases.

`HASH` overcomes this by partitioning the graph G_M (not our hypothetical G_H): a graph where each SNP column in M is a node in G_M. A SNP pair i and j are joined by an edge if at least one read fragment in M covers both i and j. Edges are weighted by the number of fragments that can connect a pair of SNPs (or if given an h, by the difference between the number of fragments that are and are not consistent with h)[19]. G_M is minimally partitioned (the *min-cut* problem [164]) to find subsets of S to add to Ψ. A cut of low weight yields a subset of columns that are inconsistent with the rest of the columns in the fragment matrix, posing a 'bottleneck' to convergence. G_M is recursively cut (*i.e.* the two subgraphs of G_M induced by the cut, are themselves cut) until only n single vertex graphs representing each of the elements in S remain. Ψ therefore contains all elements of Ψ_1, as well as subsets of S that are good candidates for flipping to improve h.

Additionally, `HASH` attempts to optimise selection of Ψ with an initial determination phase, recalculating the optimal subsets given a series of transitions made to alter h, before running the MCMC algorithm to recover a pair of haplotypes proper. First, Ψ is initialised to Ψ_1 and the MCMC is run for some number of steps ($i \approx 1000 \times n$), with each step proposing a transition (drawn from Ψ with uniform probability $\frac{1}{|\Psi|}$) to a new h that is always accepted if the probability is improved by the transition, or accepted with probability equal to the ratio of the old probability over the new if the move is suboptimal[20]. Ψ is then recalculated by cutting G_M, by reweighting the edges to consider the final h of the run. This determination phase is repeated t times (or until no improvement can be made to h in a consecutive run). The MCMC algorithm is then repeated for $j \approx 10^6 \times n$ iterations to determine the final h, initialising the starting haplotype and the list of subsets, as obtained by the last run of the determination phase (*i.e.* $h^{(0)} = h^{(t)}$ and $\Psi = \Psi_t$).

[18] A graph is strongly connected if every vertex is reachable from all other vertices.

[19] This may sound familiar, as it is the same G_M defined by Bansal *et al.* in their methodology for `HapCUT`

[20] Permitting a suboptimal transition with some probability is known as the "Metropolis update rule" [165], which when used as part of an MCMC process you might enjoy referring to as Metropolis-coupled MCMC, or MCMCMC.

The results showed that the dynamic updates applied to the MCMC approach proposed by HASH had a higher likelihood of sampling from the globally optimum haplotype solution h. HASH was fast and efficient, demonstrated to rapidly converge toward high likelihood solutions for h. The work was compared to the approach used to construct *HuRef*, showing an improvement in MEC scores over the greedy algorithm. The approach relies heavily on the quality of paired-end reads, without which thousands of disconnected haplotype blocks are created, but the authors demonstrated HASH could reconstruct human haplotypes with approximately 1.1% switch error[21]. Unfortunately for HASH, in the same year, Bansal reformulated the problem once more, and immediately superseded this work in favour of HapCUT.

2.3.5 HapCUT (2008)

Particularly of note is Bansal and Bafna's **HapCUT** heuristic [158], partly as it was one of the most popular heuristics of its time [144], but also as the work was co-authored by one of the Celera-based authors who first introduced the problem alongside Lancia in 2001. Motivated to find a less greedy approach to haplotype assembly than that introduced by Levy *et al.*, HapCUT demonstrated a significant improvement over greedy heuristics like FastHare in haplotype recovery accuracy with real data, achieving results as good as HASH in less time.

The lead author is the same as that for the previously discussed HASH approach, and defines a similar graphical problem in which each non-homozygous SNP of the matrix M is assigned a node, and SNP nodes are joined by an edge if at least one of sequenced read fragment covers both positions. The graph is termed G_M. Edges in the graph are weighted according to the difference between the number of read fragments that are inconsistent with the current pair of consensus haplotypes, and those that agree. That is, given a pair of haplotypes[22] h – described in the work as a *phasing* – an edge (i, j) in the graph $G_M(h)$ describes the "weakness" of the phasing between SNP columns i and j in M. The higher the weight between (i, j) in the graph, the weaker is the support between the haplotype pair h and the fragments that span the corresponding columns in M.

If one considers a cut C (effectively a subset of all SNPs such that $C \subset S$) of the graph $G_M(h)$, a corresponding weight for the cut $w_h(C)$ could be calculated by summing the total weight of all edges that traverse the cut and the rest of the graph $G_M(h)$. Thus, a cut C can be scored by the total fragment inconsistency introduced by the haplotype pair h, given G_M. If $w_h(C)$ is positive, then flipping the elements $h[k]$ for $k \in C$ will reduce the weakness and optimise the MEC criterion.

[21] Switch error measures the proportion of heterozygote positions whose phase is wrongly inferred relative to the previous heterozygote position [166]. *i.e.* The number of times a haplotype path "crosses" onto the path of the other haplotype, effectively removing the penalty of consecutive incorrect SNP selections, if the series is correct for the other haplotype.

[22] Although defined in the manuscript as H, I use h here to avoid the potential for misidentification of the pair of haplotypes (a set of two elements), with a set of all possible haplotypes. Although h *is* a set, warranting conventional use of uppercase notation, due to the discarded homozygous positions in M, either haplotype in a solution can be derived from the other. We could define $h = \{\eta, \bar{\eta}\}$, where $\bar{\eta}$ is the binary complement of η. Thus for later discussion we can now define a haplotype solution h as being a member of the set of all possible haplotype solutions H.

That is, for both haplotypes in h, flipping the alleles that correspond to all the SNPs included in the cut C will yield a new pair of haplotypes h_C that have a lower error given the SNP matrix M.

As a consequence, the problem of finding a pair of haplotypes h while minimising MEC is reduced to finding max-cuts in $G_M(h)$, a well described NP-hard computational problem. `HapCUT` begins with a randomly initialised h and works to refine the haplotype pairs by finding the max-cut of $G_M(h)$ and flipping the SNPs in h as necessary, until it is no longer possible to find a cut that can reduce the MEC.

The results show that `HapCUT` significantly outperformed the greedy MEC heuristics of the time, and that it was possible to obtain MEC scores as good as `HASH`, but an order of magnitude faster. The authors remarked that their previous MCMC approach held a distinct advantage of being capable of making locally sub-optimal decisions in order to try and maximise the likelihood overall, whereas a greedy approach can potentially become trapped in a local minimum. However in practice the results present a good tradeoff of being close enough (or better) to `HASH`, with a much faster running time. Given its speed and superior accuracy, `HapCUT` quickly became the *de facto* tool for SIH, but would later slowly fall out of favour due to its resource requirements when scaling to newer, larger sequencing data sets.

2.3.6 Discussion

Each of the previously discussed algorithms are specifically designed to solve the single individual haplotyping problem for humans (or at best, other diploid organisms). The presented algorithms work to locate fragments, SNPs or specific elements of the SNP matrix that are inconsistent with the solution of a pair of haplotypes. By definition the SNP matrix M only permits the formalisation of the diploid problem by restricting the elements to one of two symbols ($\{A, B\}$ or $\{0, 1\}$) or a gap $-$. The programs described pre-process the sequencing data; discarding tri- and tetrallelic sites, and ruling minor alleles with low frequency as error to ensure that M can be populated with one of two symbols only. The constraints imposed by the structure support the underlying bias that conflicts can only arise from error, and permit the discarding or altering of observed data in M until two haplotypes are reconstructed. These biases render the formalisation of the SNP matrix and the optimisations derived from it unsuitable for non-diploid (polyploid or metagenomic) analyses.

2.4 Modern Fragment Removal

Although interest in fragment removal approaches had faded in favour of the popular MEC heuristic, several later approaches went on to propose their own method that echoed the underlying idea of excluding fragments, or evidence from fragments; essentially rebranding MFR (Section 2.2.1).

2.4.1 Maximum Fragments Cut (MFC) and ReFHap (2010)

Heavily inspired by `HapCUT`, in 2010 Duitama *et al.* published **ReFHap** [167]. The work praised `HapCUT` as one of the most accurate SIH heuristics, but the authors felt a replacement was necessary upon finding it too slow to process the larger scale sequencing experiments that were beginning to become more commonplace at the time. Like `HapCUT`, `ReFHap` reduced recovering a pair of haplotypes from the SNP matrix M to that of the well-known *MAXCUT* problem. However, instead of optimising MEC, the authors proposed their own heuristic: *Maximum Fragments Cut* (**MFC**).

Before describing MFC, `ReFHap` describes a scoring function: given a pair of fragment rows r_i and r_j in the SNP matrix M, it counts the difference between the number of SNPs at which the two fragments disagree, or are the same, for all SNP columns in M (ignoring the positions where at least one fragment has a gap $-$). A larger positive score between r_i and r_j indicates many conflicting SNPs between the two fragments. MFC attempts to find cuts of rows in M such that the sum of these scores is maximised, theoretically partitioning the fragments into two sets that can be phased into a pair of haplotypes. Unsurprisingly, the work showed that MFC is also an NP-hard optimisation problem.

To reduce the problem to that of *MAXCUT*, `ReFHap` constructs a graph G, where each read fragment in M is represented by a vertex in G. Read fragments in the graph are joined with an edge if the score between the pair is non-zero. Edges are weighted by the score of its corresponding pair of fragments.

Given the edge weights, a cut of fragments (*i.e.* a selection of rows from M can be represented as the equivalent subset of the vertices in G) can be weighted as the sum of the scores obtained by pairing all the fragments in the cut, against those that would remain outside the cut in G.

The algorithm considers E, the ordered set of edges in G, by descending weight. Considering the edges with the largest weights first is arguably an improvement over random selection. One may imagine that it is likely that edges describing the largest number of conflicts between a read pair are likely to be candidates for a good cut. For a given edge $e_k = \langle r_i, r_j \rangle$, `ReFHap` builds an initial cut of G by placing the vertices r_i and r_j separately into each side of the proposed cut. While there are still vertices in G that have not yet been placed on a side of the cut, `ReFHap` finds the next vertex that adds the greatest score to the current cut, and places it on the appropriate side. Once the vertices in G have been exhausted, the cut is greedily optimised via a simple process. Considering every vertex (both in and out of the cut), if the overall cut score can be increased by swapping the position of the vertex (out of or into the cut), the flip is performed. `ReFHap` also considers flipping each of the edges along the cut themselves, whose score can be found by considering the equivalent problem of flipping a pair of vertices. The greedy improvements are iterated until the score can no longer be increased.

The entire process is repeated for the first k elements of E, each time incrementally constructing an initial cut of G given the first edge that bridges the cut, before attempting to improve with iterative local optimisations. The final cut is the one with the highest score of the k generated cuts. Although

ReFHap could consider all edges in E, the use of k is a compromise between accuracy and speed. In practice the published work uses $k = \sqrt{|E|}$, with satisfactory results.

Given a cut, M is partitioned into two sets of fragments. The final haplotype is generated effectively by taking the majority rule of the cut. For each SNP s_i column in M ($i = 1..n$), one may count the number of 0's (or A's) observed across the fragments of the cut, and the number of 1's (B's) at the same position on the fragments that are not in the cut. This yields the cut's support for the 0 allele at position i. Inversely, considering the count vice versa will yield support for the 1 allele. The symbol with the most support is selected as the haplotype allele $h[i]$. Note that ReFHap only generates one haplotype, where the other can be inferred by flipping the alleles in the returned haplotype to their opposing value; a consequence of only permitting heterozygous positions in M.

ReFHap was demonstrated to be an order of magnitude faster than HapCUT (owing to its lower overall time complexity), with results of comparable accuracy, and became one of the first popular non-MEC heuristics for SIH.

2.4.2 Minimum weighted edge removal (MWER) and HapCompass (2012)

In 2012, Aguiar and Istrail published HapCompass [168], a new graph-based method for finding the two diploid haplotypes given a SNP matrix M. The work identified that current approaches have difficulties producing accurate results, primarily as they "operate on restricted optimizations that are unrealistic considering modern high-throughput sequencing technologies".

Their method to overcome this was to introduce *minimum weighted edge removal* (**MWER**). Again, MWER could be considered as a reboot of the old MFR problem described by Lancia *et al.* in 2001. Along with the definition of MWER, HapCompass introduces a new data structure, the *compass graph*, still bound to the SNP matrix M, and is constructed in the same way as the HapCUT graph G_M (Section 2.3.5), with the exception of the edge weighting. The compass graph $G_C(M)$ is constructed such that all the SNPs (columns of the SNP matrix M) are vertices. An edge exists between SNPs i and j if at least one read fragment in M covers both SNP sites.

Consider a read fragment r that covers SNPs i and j. There are four possible combinations of the alleles 0 and 1 between $r[i]$ and $r[j]$: $\{0,1\}$, $\{1,0\}$, $\{0,0\}$ or $\{1,1\}$. The manuscript considers the cases where both alleles $r[i]$ and $r[j]$ are the same, or one is different, and uses the notation $\frac{00}{11}$ and $\frac{01}{10}$ respectfully, to refer to either of the two possibilities. Edges in $G_C(M)$ are weighted by the difference between the number of $\frac{00}{11}$ and $\frac{01}{10}$ phasings that the fragments in M, that cover columns i and j support. A positive edge provides more evidence for the $\frac{00}{11}$ phasing. The manuscript terms a non-zero edge weight as *decisive*.

The goal of MWER is to remove edges from $G_C(M)$ with minimal cost, leaving $G_C(M)$ in a state where all remaining edges are decisive, and yield a unique phasing solution for the entire graph. A

compass graph that meets these three criteria is termed a *happy* graph[23]. The work presents a method to solve MWER, that is, to make a given compass graph happy.

The algorithm begins by removing all non-decisive edges from $G_C(M)$, as they cannot contribute to the haplotype phasing, and computes T, the maximum spanning tree of $G_C(M)$[24]. Given both the compass graph $G_C(M)$ and its maximum spanning tree T, `HapCompass` enumerates through the edges $e \in G_C(M) - T$ (*i.e.* the edges that were not selected as part of the spanning tree), and reconstructs the unique cycle[25] created by reinserting each e into T by considering $T \cup e$. A cycle is marked as in conflict if it contains an odd number of negatively weighted edges, otherwise it is considered concordant.

`HapCompass` selects a random conflicting cycle and removes the edge with weight closest to 0 (*i.e.* the least decisive edge of the cycle). If the deleted edge e_{del} was in T, then the non-tree edge that connects the isolated SNP back to the tree is added to T. The set of cycles must be recomputed to consider the new T. If e_{del} was not in T, then the tree remains a spanning tree and we can select the next cycle to process. This process is iterated until $G_C(M)$ is a tree. Any spanning tree of a happy compass graph will correspond to the same reconstructed haplotype.

In practice, the published algorithm is a little more involved than the one presented in detail in the manuscript. Instead of selecting a random conflicting cycle, the step is replaced with a step that removes a set of highly conflicting edges from $G_C(M)$. As each edge can be associated with a set of cycles, one can formulate the problem of resolving multiple cycle conflicts at once[26] by finding the set of minimum weight (least decisive) edges in $G_C(M)$ that are associated with said conflicts. When an edge removal causes a SNP node to become disconnected from T, the highest weight non-tree edge is added to T to reconnect it.

The method is conceptually similar to `HapCUT` (Section 2.3.5), where maximum weight cuts are found, but instead to decide which elements of M must be flipped to satisfy the MEC criterion[27]. `HapCompass` is also similar to the concept of `ReFHap` where highly conflicting fragments are pruned out of the SNP matrix M by finding maximum cuts of a graph derived from M. `HapCompass` somewhat takes the opposing view and attempts to identify the minimum conflicting evidence in the fragments that yields a haplotype. Of course, like the other approaches surveyed, the ability to reduce the description of edge phasings to one of $\begin{smallmatrix}00\\11\end{smallmatrix}$ or $\begin{smallmatrix}01\\10\end{smallmatrix}$ is a result of only permitting recovery of heterozygous diploid organisms.

[23] Which is quite possibly my favourite piece of terminology from this entire review

[24] Given a graph G, a spanning tree is a subgraph of G that includes no cycles (*i.e.* is a tree) and connects all vertices of G. The maximum spanning tree is the spanning tree with the highest possible weight, and can be found using Kruskal's algorithm (with opposite weights).

[25] These cycles form the *cycle basis* of $G_C(M)$: the minimal set of simple cycles that can be used to describe all of the possible cycles of the compass graph.

[26] To consider the entire graph at once is too computationally expensive, thus `HapCompass` selects a subset of cycles that is first populated with the edge that is associated with the most conflicting cycles.

[27] Although, the discussion of `HapCompass` describes that it is possible to accommodate MEC with a different optimisation step, but this was more of an academic endeavour than a practical one.

HapCompass was shown to be more accurate than HapCut on real data from the 1000 Genomes Project, with fewer switch errors, with similar time and space complexity[28]. However the authors note that as a subgraph of a compass graph is also a compass graph, HapCompass is trivially parallelisable for the different components of $G_C(M)$ and can easily scale to much larger inputs than the other tested algorithms of the time. The work was well placed to become one of the most popular tools for haplotype assembly (of diploid data), showing it can scale to high-throughput sequencing data at a time when such data sets were about to become much more commonplace.

2.4.3 Balanced Optimal Partitioning (BOP) and H-BOP (2012)

A relatively uncited, but interesting formulation of SIH considers the minimisation of *both* "fragments cut" and "errors corrected", generalising both the popular MEC (Section 2.3) and recently introduced MFC (Section 2.4.1) models as the *Balanced Optimal Partition* [170] (**BOP**).

The computational implementation, a dynamic programming solution called **H-BOP**, requires two parameters, w and k. w is the weighting factor that allows the user to decide how much influence the MEC and MFC models should have over the recovery effort, with $w = 0$ corresponding to an approach that attempts to mimimise MEC, and a "sufficiently large" w corresponding to sole consideration of MFC instead. This is achieved by including w in the equations that consider the difference between the number of error corrections or fragments cut when proposing to accept new solutions. Thus H-BOP can be considered as a weighted combination of both MEC and MFC.

k is the maximum number of intermediate solutions that can be kept during each iteration of the H-BOP algorithm. When k is "large enough", H-BOP performs more like the previously described expensive exact algorithms. Unsurprisingly, the running time of the algorithm increases (linearly) with k.

I leave description and discussion of the method[29] itself outside the scope of this review, as the approach did not go on to attract much attention in future literature, though I highlight it here as an interesting considering of blending two different models together. Simulations claim that H-BOP was faster and more accurate than ReFHap, a conclusion that made its way to a review of the field by Lancia in 2016 [144], but on further inspection of the figures, they indicate less than one actual switch error difference on average between the two methods. Although the authors show that perhaps H-BOP could scale to larger data sets when compared to ReFHap and still achieve similar results, it was not enough to convince users in the field and gain popularity.

[28] The paper also chose to evaluate itself against the Genome Analysis Toolkit (GATK [169]), but found in practice the GATK required too much RAM, where in 2012 this was defined as 8 GB.

[29] Whose pseudo-code is probably more difficult to read than the implementation itself...

2.5 Modern probabilistic approaches

Early approaches to solve SIH via the popular MEC optimisation would later fail to scale with the introduction of second-generation sequencing data sets [144, 157]. `HapCUT`; arguably the most popular MEC-based heuristic, was demonstrated to scale poorly [167], paving the way in the literature to move away from MEC. Indeed Duitama *et al.* also argued that whilst heuristic approaches are not guaranteed to find the best solution in every instance, optimising the objective function of MEC did not necessarily correlate with better haplotypes, steering research away from the goal of solving such problems with exact algorithms.

Newer models began to step away from solving the described optimisations and moved to produce haplotypes with the greatest probability of being correct given the observed SNP matrix. Indeed, `HASH` was not the first probabilistic method for solving SIH, and in fact probabilistic solutions were not limited to solving the optimisations first introduced by Lancia in 2001 (and later, Lippert in 2002). In parallel independence to the approaches surveyed thus far, alternative formulations of the problem of reconstructing haplotypes from sequence data were being developed.

2.5.1 The first probabilistic formulation (Li et al. 2004)

The first such probabilistic proposal for SIH was from Li *et al.* in 2004 [171], the same year that `FastHare`: the first heuristic for MEC was published. Li *et al.* extended work from 1992 by Churchill and then co-author Waterman[30] who defined a statistical method for determining the accuracy of assembled fragments of a shotgun sequencing project, over a decade earlier in 1992[31] [172]. This earlier work introduced a statistical model which assumed the fragment assembly itself was truly correct[32], that could be used to estimate the likelihood of error per-base, or to construct a consensus sequence that satisfied a specified confidence level.

As noted earlier in Section 2.2, although unstated, it would appear the inspiration for Lancia's SNP matrix in 2001 is likely the data structure introduced here by Churchill and Waterman in 1992. Churchill and Waterman describe a matrix with m read fragment rows and n base position columns. An element in this matrix (which given the benefit of hindsight, I will denote as M), $M[i][j]$ refers to the nucleotide observed "on a gel" at position j on the i'th read fragment. Of course, the problem a decade prior considers the accuracy of only one sequence: that of the true DNA molecule that was sequenced, rather than untangling diploid sequence data to recover a solution of two distinct

[30] Who a decade later is affiliated with Celera Genomics for this 2004 work. Interestingly, the authors also thank Sorin Istrail, a co-author of the original Lancia *et al.* work in 2001, who would later co-author `HapCompass`.

[31] A time where the total nucleotides held by central databases was around 50×10^6 [172]. As of February 2018, GenBank hosts over 25×10^{10} nucleotides

[32] The model actually makes six important assumptions, including all positions and all fragments are equally reliable and that sequence error rates are uniform and independent. Interestingly, the paper cites that Markovian dependence between adjacent bases is well established [173], but claim that the effect on the final sequence is likely to be minor [172].

haplotypes. Thus, such an element was not encoded as one of two alleles, but with some symbol $\alpha \in \{A, C, G, T, -, X, \phi\}$, where X can be any ambiguously called base[33] and ϕ represents bases not covered by the sequenced fragment (in contrast to $-$, which represents a known gap in the fragment). Unlike the work surveyed so far, it is this version of the SNP matrix, rather than one containing binary alleles, that is used in the work by Li *et al*.

Of more importance to this discussion, in the same work, Churchill and Waterman introduced the *consensus distribution*; a Bayesian-based probability distribution for each position of an assembled sequence, defining the probability that a particular position had been assembled correctly given the bases observed over the fragments covering that position. It was this mathematical construct that was extended by Li *et al.* in 2004 to reformulate SIH into a statistical problem. Their objective was to evaluate $\mathbb{P}(h,\ a \mid M)$: the joint conditional probability distribution[34] of both the reconstructed haplotype pair ($h \in \{1,2\}^{2\times n}$) and the assignment of read fragments to haplotypes ($a \in \{1,2\}^m$), given data observed in the SNP matrix M. Via **Bayes' theorem**, calculating this conditional requires $\mathbb{P}(h,a,M) = \mathbb{P}(M \mid h,a) \times \mathbb{P}(a) \times \mathbb{P}(h)$ the joint distribution of h, a and M where the components represent;

- **Haplotype Frequency** $\mathbb{P}(a)$: the probability of the unknown fragment assignment vector collapses to the constant $(\frac{1}{2})^m$ as the authors fix the likelihood that one of the m read fragments belong to one of the two possible haplotypes to 0.5 and,
- **Haplotype Composition** $\mathbb{P}(h)$: the probability of the haplotype pair h is merely the product of the marginal distributions of the allele pairs selected for all n SNP positions and,
- **Sequence Accuracy (Inverse Error)** $\mathbb{P}(M \mid h,a)$: the conditional probability of observing the SNP matrix M given the haplotype h and fragment assignments a, becomes the product $\prod_{i=1}^{m} \mathbb{P}(M[i][*] \mid h = (h_1, h_2))$. *i.e.* the probability of observing the evidence over all the fragments in the SNP matrix M, given the alleles on the given haplotype pair h_1 and h_2.
  A specific[35] $\mathbb{P}(M[i][j] \mid h_k[j])$ is a fixed probability that must be specified by the user in the form of a matrix Q where an element $Q[\alpha_a][\alpha_b]$ represents the probability that the given haplotype allele $\alpha_b \in \{A, C, G, T, -\}$ is correct if the observed $M[i][j]$ was $\alpha_a \in \{A, C, G, T, -, N\}$.

Thus, Li *et al.* had proposed a statistical model based on the pair of reconstructed haplotypes, the memberships of read fragments to one of the two haplotypes, and sequencing errors. These definitions are important, as they resurface in following probabilistic works later. The aim is to find the haplotype pair h that maximises the value of this model given M. The approach has similar considerations to that of MEC, featuring a probabilistic component for determining the likelihood of sequence errors in elements of M. However, a limitation of the method is that it can only consider consecutive SNPs. To

[33] A symbol now widely replaced by 'N'.

[34] The original manuscript refers to the haplotype composition as S and the assignments of fragments to haplotypes as F. I have chosen to rename these to h and a respectively to reconcile with the notation I use later in this chapter.

[35] $k \in \{1,2\}$

minimize computational complexity, the algorithm considers single SNPs and adjacent pairs, which loses long range information between non-adjacent SNPs on the same fragment. It is noteworthy however, that unlike the majority of surveyed approaches, homozygous sites are not excluded in the words of the authors "for the sake of model flexibility and simplicity" and are not removed from the SNP matrix M, nor ignored while calculating the pairwise conditional probabilities in the second step. Additionally, the approach does not merely output a single haplotype and its complement, but maximises the allele pair at all SNP positions to reconstruct both haplotypes at the same time.

No detailed description of the algorithm is provided[36] but briefly; the algorithm begins by first computing the conditional probabilities of all allele pairs $(\alpha_a, \alpha_b) \in \{A, C, G, T, -\}^2$ at a SNP position given the corresponding column in the observed SNP matrix, for all SNP positions $j = 1..n$, across all read fragments that cover j.

Calculating all such probabilities $\mathbb{P}((h_1[j],\ h_2[j]) = (\alpha_a,\ \alpha_b) \mid M[*][j])$ for all possible combinations of alleles (α_a, α_b) and selecting the maximum allows one to find the most likely allele pair (genotype) for any individual position j. These probabilities can be stored and used in the next part of the algorithm. Typically it is not necessary to calculate all possible allele pairs as the contents of the column in M can be used to pre-filter candidates with no support.

Of course, with a single position it is not possible to determine the difference between the ordering of a heterozygous pair (*i.e.* (G,A) vs. (A,G) – the actual haplotypes) and one must extend this model to cover two or more SNPs in order to do so. The next step of algorithm is based on the same conditional as the single-loci method just described but also considers the joint probabilities of additional allele pairs[37].

Although this extended conditional could be used to consider allele pairs at many different loci, in practice the number of calculations required grows exponentially with the number of SNP sites to consider. Even though many combinations can be pruned with the evidence in M, this can quickly become intractable even for a handful of SNPs. Thus the authors consider the problem of pairs (and occasionally, triples), evaluating the probability of the conditional over all pairs of adjacent SNPs $(j-1,\ j)$ and $(j,\ j+1)$ for $j = 1..(n-1)$, for all possible allele pairs.

A finishing step attempts to reconcile the maximum likelihood alleles between adjacent heterozygous sites, from both directions (*i.e.* start to end, end to start). Where the forward $(j, j+1)$ and reverse $(j-1, j)$ reconstructions disagree on a phasing at j, the authors turn to calculating the expensive probabilities of the three allele pairs over $(j-1,\ j,\ j+1)$ to determine the correct haplotypes. If the inconsistency cannot be resolved, the haplotype is broken into two segments at j, which is left unphased. The decision between the two top phasings is settled with both a confidence and odds ratio threshold, set by the user. Homozygous sites are ignored during this particular step, as they cannot

[36] Though the discussion features the amusing phrase "implemented in programming".

[37] *e.g.* $\mathbb{P}(\ (h_1[j],\ h_2[j]) = (\alpha_a,\ \alpha_b),\ (h_1[k],\ h_2[k]) = (\alpha_c,\ \alpha_d)), \ldots \mid M[*][j], M[*][k], \ldots)$

contribute to the haplotype information. For single SNP loci that cannot be reconciled, the allele with the highest likelihood from the first step is used.

Finally, the algorithm calculates confidence scores for each of the reconstructed haplotype segments with the joint probability $\mathbb{P}(M[*][j:j+l],\ h[j:j+l])$ of both the SNP matrix M and the reconstructed segment (where j is the start of the reconstructed segment, and l is the length of the segment). This can be obtained using the product of the relevant single-loci probabilities calculated at the start of the algorithm and the marginal of corresponding columns of the SNP matrix itself. Calculating the confidence becomes increasingly complex as the length of the segment increases.

The tabular results presented show that tweaking the odds ratio and confidence thresholding (used to decide between the top two phasings from the second step) yields robust results. However, overall, the algorithm struggles in areas of low coverage and can only phase 70% of the adjacent pairs across all their simulations. Though, it correctly reconstructs around 94% of the positions that it is capable of phasing. The results of the approach are not explained in detail, and although the manuscript presents the first probabilistic method for SIH, it neglects to compare itself to any of the non-statistical alternatives of its time. However, this is the work that showed that it was possible to formulate SIH probabilistically and showed that perhaps **one of the most useful properties of a probabilistic model was that one could calculate a confidence score on reconstructed haplotypes**.

2.5.2 The first Markov chain model (Wang et al. 2006)

A later work published by Wang *et al.* in conference proceedings via the *17th International Conference on Genome Informatics* [174] introduced a new formulation of the SIH problem, that uses a Markov chain (see Section 2.3.4) to iteratively reconstruct the haplotype h from start to end, maximising the probability of the next SNP in the chain considering d ($d = 1..3$) previous SNPs.

Like the preceding work of Li *et al.*, the goal is to reconstruct the *most likely* pair of haplotypes, given a set of aligned read fragments represented by the SNP matrix M. Oddly, the approach also required the availability of a known set of unordered genotypes (that is, for each position, the called alleles). Yet, as the manuscript describes that all positions of the haplotype to be recovered must be heterozygous, it is unclear exactly what the advantage of an unordered set of $\{0,1\}$ pairs would be to the approach, though it appears that the genotype is likely used as a constraint for the construction of the states that the Markov chain can move between.

It should be noted that whilst this was the first application of Markov chains to SIH, it was not the first demonstrated application of Markov chains in a haplotyping scenario. The parallel problem of finding a haplotype that satisfies an observed population had previously been investigated with the use of Markov chains [175, 176]. This pre-existing work demonstrating the utility of the approach was likely to have inspired Wang *et al.* to apply the idea here.

The advantages of the method are weakly described. Wang *et al.* argued that if there is no prior knowledge about the data in question, then it is difficult to for one to select between the use of MFR, MSR or MEC. Yet at the time, the community had effectively abandoned MSR and MFR in favour of MEC. Additionally, the article argues that their method scales better than MEC, but unfairly compared their dynamic programming method to a computationally expensive and out-of-date branch and bound algorithm for MEC, rather than newer heuristic approaches like `FastHare` (Section 2.3.1). The work is not experimentally validated with read fragments, but instead the authors generated synthetic "SNP fragments" by copying consecutive runs of SNPs from groups of known haplotypes, with some degree of intentional error. However, the number of SNP fragments generated is very low (just 230 for their first data set consisting of 139 haplotypes), and I would have argued that these evaluation data sets were too small to support the conclusions found in the manuscript.

The results suggested **a Markov chain could achieve accuracy comparable to MEC approaches**. But counterintuitively to what one might expect (and to my work I will present later), the paper indicates that increasing d, the memory[38] of the process (from $d = 1..3$) actually *reduces* accuracy. It is unclear as to why this is the case, and no theories are presented by the authors. As the state transition probabilities ultimately depend on the observed frequencies of runs of length d across the row fragments in the SNP matrix M, it is possible that given the low number of generated SNP fragments that perhaps there were too many low probability transitions to support some of the haplotypes in the test set. Unfortunately the work concluded that Markov chains with memory offer poorer performance, somewhat stifling research in this direction for a short time.

Frustratingly, the work has appeared poorly described in several reviews of the problem of SIH, notably in 2016 with Lancia's own retrospective [144] which featured no analysis of the work and only quoted part of the abstract verbatim, suggesting the community at large did not entirely understand the contents of the work.

2.5.3 The first probabilistically reconstructed diploid sequence (2007)

In 2007, the same year as the release of *HuRef* (the first human diploid sequence; Section 2.3.2), a familiar set of names: Kim, Waterman and Li (the three authors of the original probabilistic formulation of SIH; Section 2.5.1), reconstructed the diploid sequence of *Ciona intestinalis* [177], representing the first whole-genome diploid sequence to be reconstructed with a probabilistic formulation of SIH.

Kim *et al.* apply the methodology they introduced a few years prior, first constructing their SNP matrix M where elements $M[i][j]$ are the called nucleotides themselves, rather than binary representations of alleles (M also permits gaps between mate pairs $-$, uncalled bases N, and null bases where sequencing did not occur ϕ). Recall the sequence accuracy component was originally a set of constant error probabilities, Kim *et al.* updated their formulation to instead consider position-specific `phred` [178]

[38] Here, d describes the "order" of the Markov chain, where the future state of the chain depends on the d previous states.

quality scores. Provided in the form of an $m \times n$ matrix, denoted Q such that each element $Q[i][j]$ denotes the likelihood that the base at SNP j of read fragment i was sequenced in error.

Kim *et al.* recognise the additional problem of working without a reference of polymorphic sites and that identifying SNPs *de novo* is a related problem to that of the haplotyping itself: **acknowledging that the number of possible haplotypes grows exponentially with the number of SNP sites**. Thus Kim *et al.*'s method considers the balancing of two tradeoffs between the sensitivity and specificity of both polymorphism identification[39], and haplotype reconstruction.

Potential polymorphisms are identified with Bayesian model selection[40]. Each position along the aligned reads is screened under two models: j has one haplotype, or two. The models consider the sum of probabilities for each of the possible allele homozygous singletons or heterozygous pairs at j. The ratio of support for the two models is calculated and a pair of thresholds balance the sensitivity and specificity of the approach. Sites whose ratio meets the lower threshold are categorised as *weakly* potential polymorphisms, and sites whose ratio meets the second much larger threshold are *strongly* potential polymorphisms. Other sites are assumed to be homozygous and their columns are discarded before the construction of the SNP matrix M.

Given a SNP matrix, Kim *et al.*'s approach considers three steps: local haplotyping, global haplotyping and haplotype bridging. The first step constructs blocks of four strongly potential polymorphisms (permitting weak polymorphisms to appear inbetween). Each block has an overlap of two strongly potential polymorphisms between one another. Phasing[41] is performed with a two-step Markov chain Monte Carlo approach. The first step considers each element $h[k][j]^t$ where[42] $k = 1, 2,\ j = 1..n$ in the $2 \times n$ output haplotype vector, and draws[43] a new $h[k][j]^{t+1}$ from the distribution $\mathbb{P}(h_{kj}^{t+1} \mid h_{-(kj)}^{t},\ a^t\ , M)$: the conditional distribution of observing any new haplotype configuration for position $h[k][j]$, given the current haplotype configuration $\{h_1^{t+1}, ..., h_{j-1}^{t+1}, h_{j+1}^{t}, ..., h_n^t\}$ (with the exception of position j, with notation $h_{-(kj)}$), the current fragment assignments a^t and the SNP matrix M.

Secondly, each element a_i of the read fragment haplotype assignment vector for $i = 1..m$ is redrawn from the distribution $\mathbb{P}(a_i^{t+1} \mid a_{-i}^{t},\ h^{t+1},\ M)$: the conditional probability of observing the i'th read fragment's haplotype assignment, given the other fragment assignments $\{a_1^{t+1}, ..., a_{i-1}^{t+1}, a_{i+1}^{t}, ..., a_m^t\}$ (excluding i, with notation a_{-i}), the new haplotype configuration h^{t+1} found via the previous step, and the SNP matrix M. Both conditionals collapse to a product of sequence accuracy (or error) conditionals as previously defined by the Li *et al.* manuscript (Section 2.5.1). This iterative MCMC

[39] Considering SNPs and multibase substitutions, insertions or deletions

[40] A Bayesian formulation of hypothesis testing, where the likelihood ratio of two competing hypotheses is calculated and tested to determine whether there is sufficient support for one model over another.

[41] Recall that "phasing" refers to estimation of haplotypes from genotype data.

[42] I believe there may be a misprint in the notation of the original manuscript, I have corrected the notation here such that it appears to be correct in the context of the later equations in their Methods.

[43] The underlying sampling process is a *Gibbs sampler*. This MCMC sampling approach is used when one wishes to sample from infeasibly calculable probability distributions, and here allows the configurations of h and a to converge to a highly likely solution.

process effectively updates the haplotypes given the fragment assignments, then re-updates the fragment assignments given the new haplotypes; allowing sampling of the haplotype space, guided by the fragment assignments. If the confidence score of a block meets a threshold (0.9), the block is considered to be phased, and is used as a local haplotype for the haplotype extension step.

Haplotype extension attempts to combine adjacent local haplotypes, using their overlapping strong polymorphisms. Local haplotypes where strong polymorphisms have conflicting phasing are extended to the closest next strong position, and the MCMC sampling step is reapplied to the extended block to try to correct a potentially bad phasing. If shared strong positions still feature alternate phasings, the problem is considered from the forward and reverse directions and the majority support for the phase is chosen. If the confidence of this position is low, the haplotype block is split and the position is ignored. A final bridging step defined in the supplement describes how extended haplotype blocks that cannot be extended further due to not meeting a confidence threshold, can potentially be linked together with fragment information directly. Similarly to the identification of polymorphic sites at the beginning of the process, model selection is used to select between alternate hypotheses that a subset of read fragments support bridging a particular pair of extended haplotypes. Once extension possibilities are exhausted, the approach terminates, and the state of the haplotype blocks is the solution returned.

Kim *et al.*'s approach **demonstrated the first large-scale application of a probabilistic formulation of SIH**. *Ciona intestinalis* was selected as an ideal candidate given its high rate of polymorphism (estimated to be 1.2%), and the method was able to recover thousands of uninterrupted diploid gene sequences for further analysis. The work notes its reliance on suitably sized reads (average 550 bp), the sensitivity to error, the requirement of a good assembly and the need for enough polymorphisms to be identified in the first step. The selection of appropriate values for the two thresholds to determine weak and strong SNP positions is left to the user.

Interestingly, as part of the Discussion, the authors note that their work is capable of considering more than two haplotypes, which may be one of the earliest *considerations*[44] of the computational problem of polyploid recovery from sequence reads. Kim *et al.* note their model selection and MCMC methodology could be extended to consider a fixed number of haplotypes above 2. Though, the extension is not considered in detail, and arguably this formulation would add a significant number of additional sampling steps that could have precluded the method from operating efficiently. Amusingly, the authors do note that their "focus on reconstructing [diploid haplotypes], avoid[ed] the issue of assembling polymorphic genomes". Like the work of Levy *et al.*, it does not appear that the diploid reconstruction method was named or made generally available, preventing further or wider evaluation or use by the community.

[44] ...which is very different from an actual *formulation*...

2.5.4 MixSIH (2013)

Bridging a gap across these earlier works, a 2013 article by Matsumoto and Kiryu noted that few in the field had attempted to form the problem of SIH probabilistically, highlighting the previously described work of Li *et al.* in 2004, recognising that the complexity grew exponentially with the number of SNPs, requiring the approach to only consider neighbouring SNP sites. Additionally, the confidence scores must be calculated over all possible haplotypes, an expensive operation as the length of the reconstructed sequence increases. To overcome this, Matsumoto and Kiryu proposed **MixSIH** [179]: a novel probabilistic approach, which considers read fragments as independent observations, rather than Li *et al.* who correlated fragments together with hidden states that represented the true haplotype. That is, the fragment assignment vector (that we termed a in Section 2.5.1 and will continue to do so here[45]) is no longer represented by a probability distribution related to all fragments. Each read fragment r_i ($i = 1..m$) has a corresponding $\Phi^{(i)}$ that represents a partially recovered haplotype over the SNPs covered by r_i.

The supplement of `MixSIH` acknowledges that Li *et al.*'s consideration of fragment emission as correlated is more natural (as fragments *must* originate from one of the two haplotypes), but this contributes to the model's significant complexity, as a solution must consider all possible combinations of the read fragments assignment vector a. `MixSIH`'s approach exchanges the concept of each read fragment being connected to a common hidden haplotype state, with the individual haplotype states per read $\Phi^{(i)}$, which loses complex inter-fragment relationships, but drastically reduces the complexity of calculating likelihoods, allowing `MixSIH` to scale to larger data sets.

`MixSIH` still uses the SNP matrix M, but restricts an element $M[i][j]$ to be an allele $\in \{0,1\}$ or otherwise $\varnothing$, if the position is gapped, unaligned, ambiguous ('N') or different from the two possible alleles for j. The work defines a mixture model[46] for the joint probability of the fragment haplotype assignments a, the collection of partial haplotypes $\Psi = \{\Phi^{(1)}..\Phi^{(m)}\}$ and the SNP matrix M itself. Given a set of parameters Θ, `MixSIH` considers this joint distribution over all read fragments: $\mathbb{P}(M[i][*],\ \Phi^{(i)},\ a[i] \mid \Theta) = \mathbb{P}(M[i][*] \mid a[i],\ \Phi^{(i)}) \times \mathbb{P}(a[i]) \times \mathbb{P}(\Psi^{(i)})$.

This joint distribution is similar in formulation to that of Li *et al.*, requiring the calculation of the same three distributions described in Section 2.5.1. Familiarly, `MixSIH` too defines the haplotype emission distribution $\mathbb{P}(a[i] = 0) = \mathbb{P}(a[i] = 1)$ as a fixed constant of 0.5, and the sequence accuracy conditional $\mathbb{P}(M[i][*] \mid a[i], \Psi^{(i)})$ is similarly defined as the product $\prod_{j, r_i[j] \neq -} \mathbb{P}(M[i][j] \mid \Psi^{(i)}_{a[i]}[j])$, where j enumerates the SNP sites covered by read fragment r_i. This conditional defines the probability of observing the allele at $M[i][j]$, given the allele at SNP j of the partially reconstructed haplotype $\Psi^{(i)}[j] = a[i]$ (where $a[i]$ is 0 or 1). This conditional is simply set to be $1 - \varepsilon$ if the two alleles are the

[45] Confusingly, the fragment assignment vector is referred to as h in the `MixSIH` manuscript, where every other work including this review uses h to refer to the haplotype itself. For reasons that will become clearer, Φ is used to refer to the haplotype "phase vector" in this work.

[46] Essentially, a distribution containing a mix of component distributions[180]. Here, the mixture components represent the two haplotypes.

same, and ε otherwise. For their evaluation, ε is set at 0.1, although it could be estimated from the SNP matrix M or other sources if desired. Finally, the marginal probabilities of $\mathbb{P}(\Phi^{(i)})$ are calculated as the product $\prod_{j,r_i[j]\neq -} \mathbb{P}((\alpha_a,\alpha_b)_{(j)})$: the probability that position j yields the phased genotype (α_a,α_b).

The parameter set Θ is responsible for defining these per-position genotype probabilities. *Variational Bayesian Expectational Maximization*[47] inference (VBEM) is used to optimise Θ. Whilst the details of this are outside the scope of this review, in brief, the use of VBEM here attempts to minimise the divergence[48] between two probability distributions $P(a,\Psi,\Theta \mid M)$ and $Q(a,\Psi,\Theta)$, essentially minimising the information lost by knowing about the SNP matrix M[49]. The VBEM method is iterated until the distribution converges (altering the parameter configurations to try and avoid local minima), allowing the selection of maximum likelihood per-position phased genotypes $\mathbb{P}((\alpha_a,\alpha_b)_{(j)})$. The manuscript provides insufficient detail on how the components of their approach (the statistical mixture model and VBEM parameter estimation) fit together to actually produce haplotypes.

The authors recognise that their model could be improved by considering the actual nucleotide phases (*e.g.* $\{(A,C),(A,G),...(T,C),(T,G)\}$), rather than the binary phases $\{(0,1),(1,0)\}$. Thus considering the genotypes simultaneously, as later achieved by `ParticleHap` (Section 2.5.6). One could even consider homozygous sites, small insertions or deletions by adding them to the set of possible phases. `ParticleHap`'s Ahn and Vikalo also identified that the sequence accuracy conditional probability could instead be dependent on the fragment matrix or some other more accurate measure than the fixed ε.

`MixSIH` also introduces several accuracy measures that are not used by future literature. The work presents results on both simulated and real data, although the competition appears to achieve much worse results in simulation than on real data, perhaps indicating a poor choice of mock data. The model is sensitive to selecting an appropriate threshold for their "minimum connectivity" score that determines whether a reconstructed haplotype should be broken into two segments at a given position. The results focus on the relationship of this score, rather than on a comparison between `MixSIH` and its contemporaries. It is unclear whether there is an advantage in using `MixSIH` over `HapCUT` or `ReFHap` which appear to perform more favourably under some circumstances in their presented figure. Indeed, in a discussion on running time, Matsumoto and Kiryu acknowledge that the repeated application of VBEM is computationally expensive, and show `MixSIH` is an order of magnitude slower than heuristic alternatives like `HapCUT`, which ultimately seems to be responsible for the lack of uptake of the presented work.

[47] Expectation-Maximization (EM) is an iterative process to estimate the value of parameters in a statistical model, with the maximum likelihood.

[48] Kullback-Leibler (KL) divergence can be referred to as *relative entropy*, is a non-symmetric measure of the difference between two distributions P and Q. The KL divergence of Q to P (denoted $D_{KL}(P\|Q)$) measures the expectation of the log difference between P over Q. One may consider this as a measure of information lost when using Q to approximate P [181]

[49] The manuscript describes $D_{KL}(Q\|P)$, contrary to textbook examples, implying they wish to measure and minimise the divergence of P to Q instead: the information lost by considering the SNP matrix M in addition to the rest of the parameters.

2.5.5 ProbHap (2014)

In 2014, Kuleshov discussed that SIH still posed a difficult, unresolved computational problem, largely owing to evidence provided by short-read sequencing technologies, but recent advances in fosmid-based preparations had made the sequencing of longer DNA fragments more readily available. Such long reads[50] can provide much more information to algorithms attempting to solve SIH by offering more overlapping positions that can be used to more easily determine the true origin of the fragment. Kuleshov suggests that SIH has become more tractable with the introduction of these longer sequencing technologies, but handling long read data "remains non-trivial computationally". Kuleshov acknowledges that no such formulation of the problem so far can work with such reads and introduced **ProbHap** [182] to fill the niche.

`ProbHap` defines a probability function $\mathbb{P}(h, a, M)$ whose value depends on the observed SNP matrix M[51], the unobserved true first haplotype (defined by the vector $h \in \{0,1\}^n$), and the unobserved true fragment assignments (defined by the vector $a \in \{0,1\}^m$). This will seem familiar if you recall the definitions used in the previously discussed 2004 work by Li *et al.* (Section 2.5.1), and the probability function is the *same* joint probability function introduced earlier. Note though, in this instance, the h vector encodes just one of the two haplotypes.

The joint probability is formally defined in this work as a product of factors, including the conditional probability $\mathbb{P}(o_{ij} \mid a[i],\ h[j])$: the probability of observing the allele given by $M[i][j]$ (the j'th SNP on i'th read fragment), referred to by Li *et al.* as the sequence accuracy component of the joint distribution. In Kuleshov's definition, the conditional probabilities are found by looking at the corresponding base quality score $Q[i][j]$, in the $m \times n$ quality matrix Q. That is, a read's contribution depends on its per-base quality scores. Quality data was more readily available in 2014 with modern sequencing platforms, whereas the work by Li *et al.* a decade previously, relied on a user-provided set of probabilities that defined the likelihood of observing a particular allele (independent of read fragment or SNP position), given an allele assumed to be correct.

The conditional is referred to in the rest of the manuscript as P, and can be represented graphically: given the observed SNP matrix M, one can consider a graph whereby all elements $M[i][j] \in M$ are connected to a hidden row state indicating the true fragment assignment (a_i), and a hidden column state indicating the true haplotype allele (h_j). `ProbHap` aims to determine h^*, the haplotype that maximises the probability of the observed read fragments occurring as described by the SNP matrix M.

[50] In the context of this work, long reads here refer to fosmid-based technologies, rather than the PacBio and Oxford Nanopore methods that we colloquially refer to as long reads technologies today

[51] Confusingly, the manuscript flips the meaning of m and n as established by Lancia and used throughout the surveyed literature. To avoid confusion, I continue to use the definitions from Section 2.2. Additionally, I rename the vector used to represent the assignment of reads to haplotype 0 or 1 to a (from r) to avoid conflict with the use of r as a read fragment $\in R$.

ProbHap achieves this with *belief propagation*, but first needs to convert M into a suitable data structure. Briefly[52], one may consider the product of the conditional P, as a product of n SNP position factors ($\phi_1 ... \phi_n$). Each ϕ_i is a function defining a relationship between a subset C_i that contains the read fragments[53] $r \in R$ that cover SNP position i. Graphically, clusters can be arranged into a *cluster graph*, and as each cluster of reads C_i pertains to a SNP, they can be ordered by $i = 1..n$, thus the *cluster graph* is also a linear graph (a tree), T. Finally, as a read is a member of all C_i for each SNP position i which it spans, T must be a *clique tree*[54], and a valid tree for representing the function P.

As T is a clique tree, it is now a structure amenable to the use of *belief propagation*[55], which efficiently computes all the marginal probabilities of the hidden states describing the haplotype vector and read assignment vector, by passing messages towards the tree root, describing the probabilities of the edges between nodes in the tree. For the sake of brevity, a full description of the belief propagation algorithm itself is perhaps better found in a suitable textbook [181, 184].

ProbHap performs favourably, producing haplotypes with more accurate long-range reconstructions, and lower switch error rates when compared to the best competitor of the time, ReFHap (Section 2.4.1). The work was also compared against FastHare, despite being a decade old at this point, as well as HapCUT and MixSIH – but no mention of HapCompass. One may consider Kuleshov's work as an independent comparison.

With regards to the formulation of the probability distributions, the work appears to have reinvented the wheel unintentionally, as no there is no reference to the Li *et al.* work in the manuscript. However, the application of belief propagation to T is novel, considerably more efficient and reveals more probabilistic information that can be used to reconstruct h. Note that this work still scales poorly with coverage, as the number of messages to pass around T before the structure can answer questions about the likelihoods of h becomes too high. In its worse case, computational complexity increases exponentially with read coverage. The manuscript is clear that the tool was specifically designed for low-depth long-read data, for data sets with up to 10-12$\times$ coverage (although it does offer a naive fragment merging mechanism to artificially reduce the depth of a sample). Indeed, the manuscript explicitly stated that ProbHap is inappropriate for high coverage short read data.

Kuleshov only credits MixSIH with the ability to return confidence scores for haplotypes, although we have seen that the ability to do so was first formulated by Li *et al.* in 2004. Kuleshov's method does however take the concept of returning confidence scores further, and ProbHap offers three

[52] A function P with a set of n variables $x \in X$ can alternatively be represented (*factorized*) as a product of some number of factors, where a factor is a function defining a relationship (such as a constraint) between a subset of the variables in X. One may depict such a factorisation with a *factor graph* in which functions are denoted with square nodes, and variables as circles. Edges connect a variable x_n to a function f_m if $f_m(x_m)$ has dependence on x_n [181].

[53] Strictly speaking, C_j contains the unknown single haplotype allele $h[j]$ and both the unknown fragment assignments $a[i]$ and the actually observed alleles $o_{i,j} = M[i][j]$ for all read fragments i that span the SNP position j. However, any intersection between a given pair of $c \in C$ will only yield elements from the fragment vector a.

[54] Essentially, a *clique tree* (or *junction tree*) is a cluster tree that satisfies the *running intersection* property: where if a variable x is in C_i and C_k then it must also appear in any clusters C_j ($i < j < k$) appearing in the path between C_i, C_k [183]

[55] Referred to as *max-sum message passing* in the manuscript

different probabilistic metrics for each haplotype 'block' (a reconstructed region). Of course, suitable thresholds for these three parameters must be provided by the user.

- **posterior**: the probability of correctly recovering a SNP in a block, given the first SNP of the haplotype block, an indicator of the quality of the phased block
- **transition**: the probability of correctly recovering a SNP, given the previous SNP, if the transition probability is low, this may indicate that the haplotype block should be split
- **emission**: the product of the conditional P (the sequence accuracy component of the Li *et al.* model), which can be useful to find sequencing errors across fragments, indicating SNPs where you may want to ignore the phasing entirely

Despite echoing many results from Li *et al.*'s ten year old work, the novel application of a modern computational technique to reduce the problem of calculating all those probabilities makes the problem somewhat more tractable (albeit over small coverage sequencing). The additional metrics afforded by the framework made `ProbHap` a popular alternative to `ReFHap` for suitable data sets.

Perhaps most interestingly for our story, after the discussion of `ProbHap`, the paper goes on to reformulate the existing well-known problem of MEC as a special case of Kuleshov's probabilistic framework. This not only allows exact solutions for the MEC criterion to be generated, but additionally makes it possible to produce actual probabilistic confidence scores for the reconstructed haplotype blocks too. Kuleshov concludes that this result could form a foundation for further theoretical insights to SIH. Unfortunately the work appears to have been poorly cited in the literature[56], featuring only in reviews only to reference `ProbHap` itself, perhaps arriving a too late to join the previous decade's more theoretical insights on SIH, or to revive interest in the MEC optimisation.

2.5.6 ParticleHap (2015)

In 2015, Ahn and Vikalo surveyed and recognised the ubiquity of the SNP matrix M across the different formulations of the SIH problem [186]. These formulations focus only on biallelic SNPs, requiring discarding or altering of information observed to first construct M in a manner that permits only one of two alleles $\in \{0,1\}$. Ahn and Vikalo argued that this property, combined with the focus on the literature at the time on modelling sequencing errors in the SNP matrix M was a weakness, and that **trusting the genotypes used to construct a binary M could lead to incorrect haplotype reconstruction**. Ahn and Vikalo cite the low-coverage regions from the 1000 Genomes Project pilot which demonstrated erroneous heterozygous genotypes, even for high frequency variants [187].

Discussing probabilistic works in particular, Ahn and Vikalo identified `MixSIH` as the state-of-the-art for probabilistic reconstruction at the time, but noted that it is at least 10-fold slower to compute than

[56] Perhaps explained by the tongue-in-cheek opinion of Fawcett and Higginson that the number of equations in a manuscript negatively correlates with its citation rate [185].

popular heuristics such as `HapCUT` (MEC, Section 2.3.5) or `ReFHap` (MFC, Section 2.4.1). Thus, Ahn and Vikalo introduce **ParticleHap** [186], a novel, efficient probabilistic method, that incorporates genotype calling during haplotype reconstruction. The approach uses the same notation as Li *et al.*'s SNP matrix (Section 2.5.1), and so `ParticleHap` does not consider a binary representation of the SNPs, but the original called nucleotides themselves. Thus this construction of the SNP matrix M permits SNP columns to contain more than just two different nucleotides (and the gap symbol $-$).

The goal is to determine the maximum likelihood pair of haplotypes $h = (h_1, h_2)$, given the SNP matrix M, to maximise $\mathbb{P}(h \mid M)$. Again, the conditional probability $\mathbb{P}(h,\ a,\ \mid M)$ can be calculated given the familiar joint probability (Section 2.5.1): $\mathbb{P}(h,\ a,\ M)$ over $\mathbb{P}(M)$. As before, the fragments are assumed to be generated from one of the two haplotypes with fixed probability of 0.5, thus collapsing the haplotype frequency distribution to the constant $(\frac{1}{2})^m$. Like `MixSIH`, the sequence accuracy component is a fixed lookup of sequencing an allele α_a, given the haplotype is assumed to be α_b, and is assumed to have an error of $\varepsilon = \frac{0.01}{3}$ if the alleles are different, and a probability of 0.99 if they are the same.

As we have seen, finding an exact solution for this conditional distribution is computationally difficult. `ParticleHap` considers a system with initial state $\mathbb{P}(h[1])$, transitions between successive states $\mathbb{P}(h[t] \mid h[t-1])$ and state measurements $\mathbb{P}(h[t] \mid M[*][t])$. `ParticleHap` employs a *particle filter*[57], to sequentially reconstruct the most probable posteriori probability distribution $\mathbb{P}(h_j \mid M[*][j])$ for each SNP position $j = 1..n$, by modelling the potential haplotype states as weighted particles, where the weights reflect the probability of sampling that state, and recursively depend on the weighting of the particle's previous state. The system sequentially propagates transition probability values around the system as weights, to gain a suitable estimate for the conditional $\mathbb{P}(h \mid M)$, and thus a haplotype solution h.

The algorithm additionally depends on two variables: K, the number of particles to consider per SNP position[58], and L, the number of possible states that can be reached by extending any current haplotype state – *i.e.* the 12 different heterozygous pairwise combinations of $\{A, C, G, T\}$. The article notes that L could be reduced in practice, given the observed data in the SNP matrix M.

The algorithm is not sufficiently explained in the manuscript. The first step considers initialisation of the initial system state $\mathbb{P}(h[1])$, computing the set of L possible weights $w^l_{j=1} \propto \mathbb{P}(M[*][1] \mid h_1^{(l)}[1] = \alpha_a,\ h_2^{(l)}[1] = \alpha_b)$ $(\alpha_a \neq \alpha_b)$ for the L possible genotype states at $M[*][1]$. The second step, considers each SNP $t = 2..n$ in turn, for which one can enumerate the L possible extensions (appending an allele state) of the current set of particles (which represent a sample of the current possible haplotype reconstructions from SNP $1..t$), generating new weights for each particle state. During this step, any non-zero weight can be added to the particle set, until K particles have been generated. Once K

[57] Alternatively referred to by the authors as *sequential Monte Carlo*

[58] Which on a check of the source, appears to be decided by `ParticleHap` to be between $12 - 50$ based crudely on the number of heterozygous sites observed in the data.

particles exist, new particles may be swapped into the set if the weight is greater than at least one of the particles in the set. When $t = n$, one may select the particle with the greatest weight: yielding the most likely haplotype state h. Interestingly, to improve accuracy, the algorithm is executed twice: in both the forward and reverse directions.

It is unclear exactly how the algorithm works; such as how the K particles are populated for initialisation and updated, or the mechanism that reconstructs h given the most probable particle of the set, and viewing the source yields no obvious further assistance[59]. Despite this, the results presented show that `ParticleHap` is capable of reconstructing haplotypes with more accuracy than the non-probabilistic alternatives `ReFHap` and `HapCUT`, with slightly faster running time. Thus providing a probabilistic alternative to `MixSIH`, that can compete in running time with state-of-the-art heuristics. Curiously, while Ahn and Vikalo identified `MixSIH` as the current state-of-the-art for probabilistic reconstruction, they do not compare their approach against it.

The work is important as it **shows the usefulness of a 2^{nd} order Markov model**[60]; considering both the current t and previous $t-1$ states when attempting to reconstruct $t+1$. Additionally, `ParticleHap` was able to phase an extra $1.2-1.5\%$ SNP sites when compared to `HapCUT` and `ReFHap`, yielding **more accurate haplotypes from considering the actual nucleotide sequences of the fragments in M**, rather than the binary formulation that discards information from tri- and tetra-allelic sites.

2.5.7 HapCUT2 (2017)

More recently, Edge, together with Bafna and Bansal extended `HapCUT` in the form of **HapCUT2** [188]. The work highlights the ongoing desire of the field to reconstruct haplotypes of whole-genome sequences, noting that the difficulty of doing so computationally has led to a number of new sequencing protocols that attempt to maintain the integrity of as much of the haplotype information as possible to ease the problem of SIH post-sequencing. The work of Edge *et al.* considers the metaproblem of poor performance from SIH algorithms when given data from newer sequencing technologies and extends `HapCUT` to handle multiple such technologies that were previously unconsidered and unsupported by the state-of-the-art including; dilution pool sequencing, linked-read sequencing, single molecule real-time (SMRT) sequencing and proximity ligation (Hi-C) sequencing.

`HapCUT2` still uses the familiar SNP matrix M, and still uses binary strings of 0 and 1 to represent the alleles (and the gap $-$) observed at heterozygous SNP sites across sequenced read fragments. Recognising the versatility of probabilistic approaches (Li *et al.*, Section 2.5.1; `HASH`, Section 2.3.4; `ProbHap`, Section 2.5.5), `HapCUT2` differentiates itself from its MEC-based predecessor and swaps out the objective function in favour of a statistical formulation of SIH.

[59] The manuscript isn't entirely clear, and so I direct further discussion on deterministic sequential Monte Carlo for the interested at either of the two PhD theses on the subject cited by the manuscript.

[60] The approach also provides a mechanism to allow fragments with gaps to still contribute to the model by considering the nearest informative (non-gap) position if $t-1$ is a gap.

Consider a haplotype vector $h = \{0,1\}^{2\times n}$, representing a haplotype pair (h_1, h_2), and a sequencing read fragment $r_i \in R$, one may define the probability of observing r_i, given h (and an $m \times n$ quality score matrix; Q) as the distribution $\mathbb{P}(M[i][*] \mid Q[i][*], h)$. This distribution can be nicely calculated as the product of observing each of the non-gapped alleles on the fragment r_i: $\prod_{j,r_i[j]\neq -} \frac{\mathbb{P}(M[i][j]=h_1[j])+\mathbb{P}(M[i][j]=h_2[j])}{2}$. The individual probabilities that a particular $M[i][j] = \alpha$ is trivially defined as the quality probability $Q[i][j]$ if $M[i][j] = \alpha$ and $1 - Q[i][j]$ otherwise. Thus, one can now compute $\mathbb{P}(M \mid Q,\ h)$ as a product of this read likelihood function, over all read fragments: $\prod_{i=1}^{m} \mathbb{P}(M[i][*] \mid Q[i][*], h)$. Additionally, the authors must define a likelihood function: $\mathscr{L}(v) = log(\ \mathbb{P}_C(M \mid Q,\ h^*(C_1 \cup v))\ ) - log(\ \mathbb{P}_C(M \mid Q,\ h^*(C_1))\ )$, which describes the log difference between considering haplotype h with flipped elements[61] $h[j]$ for $j \in (C_1 \cup v)$ and $j \in C_1$. A positive $\mathscr{L}(v)$ indicates a higher likelihood h can be achieved by adding SNP v to the cut C_1.

`HapCUT2` revives the concept of the "read-haplotype graph" G_M, introduced by `HapCUT` (Section 2.3.5): a graph whereby each non-homozygous SNP is assigned a node, and SNP nodes are connected by an edge if at least one sequenced read fragment covers both corresponding positions. Similar to `HapCUT`, the approach searches for a cut in the graph $G_M(h)$, whose flipped elements will yield a better solution. Unlike minimizing MEC, `HapCUT2` calculates whether a given cut will yield a higher likelihood h^* than the current haplotype configuration h.

The core algorithm is straightforward, and initialises each cut by selecting a random edge in the graph G_M, and assigns one of the two vertices $\in V$, to C_1 and C_2, respectively. While there are still SNP vertices in the graph that have not been added to a side of the cut, `HapCUT2` considers each in turn, calculating $\mathscr{L}(w)$ for $w \in V - (C_1 \cup C_2)$ and selecting the w with highest absolute value $|\mathscr{L}(w)|$. If the selected $\mathscr{L}(w) > 0$, w is added to C_1, else if the log likelihood was negative, it as added to C_2. In the case where the likelihood is not affected at all, w is added to one side of the cut at random. Once all vertices in G_M are contained by either side of the cut C_1 or C_2, `HapCUT2` calculates whether the flipped haplotype $h^*(C_1)$ increases the probability of observing M, *i.e.* $\mathbb{P}(M \mid Q,\ h^*(C_1)) > \mathbb{P}(M \mid Q,\ h)$. If it does, the haplotype configuration h is flipped as-per $h^*(C_1)$ and the maximum cut algorithm starts over with a new edge in G_M.

Maximum likelihood cut iterations continue for a fixed number of loops (default $10,000$) or until it is not possible to improve the likelihood of h^* over h for a number of consecutive loops (defaults to 5). It is worth nothing that the algorithm also considers each component of G_M separately, allowing parallel consideration of multiple haplotype blocks at once and each block has its own loop counter. In practice the starting edge is not randomly selected and an approach similar to the one found in `ReFHap` is used to order a subsample of the edges and select a starting edge with low weight ($\mathscr{L}(w)$) to initialise a good cut.

A post-processing step enumerates over the reconstructed haplotype, independently considering each genotype $(h_1[j], h_2[j])$ for $j \in 1..n$. For each position $h[j]$, the four possible phasings $\{00, 01, 10, 11\}$

[61] *i.e.* Swapping the values of $h_1[j]$ and $h_2[j]$ for all suitable j.

are reconsidered, and if the likelihood of the haplotype can be improved by overriding the reconstructed haplotype, h is updated. Additionally, for each position $h[j]$, HapCUT2 considers the maximum posterior probability: $\arg\max_{x\in\{00,10,01,11\}} \mathbb{P}(h^*[j] = x \mid Q, M, h)$. If this probability is below some user-defined threshold (defaults to 0.8), the position is pruned from the haplotype. HapCUT2 also employs a mechanism to evaluate whether a haplotype should be split into blocks at low-confidence sites.

The read likelihood $\mathbb{P}(M[i][*] \mid Q[i][*], h)$ assumes a model where a read is copied from one of the two haplotypes with some error (modelled by the quality score matrix Q). In practice this likelihood is modified to consider error rates or characteristics of a particular sequencing platform (the manuscript describes an example where the likelihood is formulated to consider the an additional parameter for the probability of a Hi-C read being *in trans* (paired to a different chromosome)).

The evaluation is extensive, comparing the approach to FastHare[62], ProbHap, ReFHap and HapCUT. Notably, DGS, MixSIH and HapTree are excluded as they did not perform sufficiently well on the test data. The work demonstrates HapCUT2's scalability is "unmatched" by any of the state-of-the-art competitors, reconstructing haplotypes with fewer errors, for PacBio, 10X and Hi-C data. The evaluation showed on a simulation of a large data set (a synthetic 250Mb chromosome) that ProbHap and ReFHap scaled poorly with coverage (as expected, Section 2.4.1; 2.5.5), that the original HapCUT quickly hit the memory limit of 8GB in most cases, and FastHare ran very quickly but with a high switch error rate[63].

HapCUT2 is designed specifically for newer long read technologies, and identified the **importance of considering the idiosyncrasies of the sequencing method used to generate the read fragments** and that **probabilistic methods provide a framework for easily including the modelling of such technology-specific characteristics**. The greedy maximum cut approach, paired with a maximum likelihood method is efficient, and the early loop breaking mechanism that stops the search for a better cut significantly reduces running time. The work concludes that future contributions to the field should further consider the capabilities of sequencing technologies for haplotyping of individuals.

2.5.8 HapChat (Preprint, 2018)

Earlier this year, Beretta and Patterson *et al.* combined forces and published a pre-print on their new approach: **HapChat**[64]: *Haplotype Assembly Coverage Handling by Adapting Thresholds* [189]. Their introduction reviews two recent works: WhatsHap [190] (2015) and HapCol [191] (2016); both of which are tools constructed by the authors of the HapChat manuscript.

[62] Despite being over a decade old!

[63] In one case, not appearing on a graph with a truncated axis as the error rate was too high.

[64] Continuing along the whimsical theme of contrived names that are homonyms of popular social media apps.

WhatsHap reintroduced a formulation of minimum error correction (Section 2.3) that had appeared briefly in the literature: *weighted MEC* [192] (**wMEC**) by considering the `phred` sequence scores as probabilities that a given base is correct to weight the objective function. `WhatsHap`'s novelty arises from the consideration of sequencing coverage as a fixed parameter, and constructing a dynamic programming solution that attempts to find the optimum bipartition of SNP matrix M to flip such that it is feasible at minimum cost. The dynamic programming algorithm considers a matrix of all possible bipartitions of the fragments at each SNP j, and calculates the cost. One can then backtrack through this matrix to reconstruct the optimal bipartitions to flip to recover the lowest cost solution. The use of fixed coverage determines the maximum number of fragments that can overlap at a particular j, limiting the number of possible bipartitions that must be computed for each j.

Having reintroduced wMEC to the literature, the authors were unaware of any alternative wMEC method that would be suitable for evaluation on longer reads, and chose to compare themselves to older dynamic programming methods with unsurprisingly positive results. `WhatsHap` is limited to data with 20× coverage, though the authors claim to have not found an improvement in accuracy with higher coverage, this was only experimented on simulated data. The authors note that future will work consider how to potentially break the problem up such that they can handle higher coverage read data.

HapCol introduces its own novel formulation of the MEC: *k-constrained MEC* (**k-cMEC**). The authors exploit the assumption that sequencing errors are uniformly distributed across reads (which is true for 'PacBio' [193], but not for other methods, such as Oxford Nanopore [194]), to constrain their MEC approach to correcting at most, k fragments per SNP. `HapCol` iteratively computes the "k-corrected matrix" (that is, the corrected SNP matrix M) by considering consecutive columns of the SNP matrix in turn, calculating all possible corrections that can be applied to columns $M[*][j], j = 1..n$ and choosing the one with minimum cost that must be below k. The algorithm can reconsider specific elements of previously corrected columns if it would lower the cost, but remain within the per-column budget of k. A recursive approach constructs the corrected M. In practice, the algorithm considers two additional parameters that can permit costs above k: the estimated substitution error rate (ε) and the probability that a column contains more than k errors (α).

`HapCol` was demonstrated to marginally outperform `WhatsHap`; phasing more positions with fewer errors. Both outperformed `ProbHap` and `ReFHap`. Simulations showed that `WhatsHap` scaled poorly with coverage, with memory usage increasing two orders of magnitude over `HapCol`'s usage, to over 128 GB of RAM at 20× coverage. However, the results also demonstrated that in several cases, `HapCol` was terminated as no feasible solution could be found with the provided α and ε. Clearly, data sets where many columns contain more errors than k pose a problem for the approach, and require tweaking of the input parameters.

Beretta and Patterson *et al.* summarise that both `WhatsHap` and `HapCol` have trouble with data sets with coverage above 20×. The two approaches however, are more suited to long reads than either `ReFHap` or `ProbHap`, backing up the later conclusions in Edge *et al.*'s evaluation of `HapCUT2`. The

authors contextualise HapChat, as a method capable of overcoming the challenge of read coverage. The approach builds on both concepts: extending HapCol to consider the modelling of per-column errors, but allowing this parameter to be adapted during the running of the algorithm; and using a similar approach to WhatsHap to merge reads that are likely to have originated from the same haplotype.

Once more, we consider the SNP matrix M, that permits an element to be one of two alleles, or a gap. Note that the formulation of M in these approaches permits homozygous columns. Both HapCol and HapChat consider each SNP column $M_{\cdot j}$ of M as a vector over all fragments $\{0,1,-\}^m$. Borrowing some notation from WhatsHap, one may define the coverage cov_j of any SNP site j as the number of active read fragments at j, and any fragment r_i with a non-gap ($\neq -$) element at j is described as *active*. HapCol defined a conflict between two SNPs positions i and j, if there exists a pair of active fragments r_u and r_v such that a 2×2 submatrix $M[u,v][i,j]$ is monomial[65]. Additionally, the function $\delta(M_{\cdot i}, M_{\cdot j})$ returns the minimum number of corrections that must be considered to make the two vectors of i and j compatible (*i.e.* to resolve the conflicts observed on active fragments). A *k-correction* of $M_{\cdot j}$ is obtained by flipping at most k active (non-gapped) elements of the corresponding column vector, while minimising $\delta(M_{\cdot i}, M_{\cdot j})$, where k is a fixed error-rate parameter provided to the model.

HapChat describes a small modification to HapCol's k-cMEC approach, defining the acceptable number of errors as a recurrence relation that depends on each previous SNP position, adding a logarithmic term at each j to permit a small degree of additional error than the defined k. The authors term this "adaptive k-cMEC", a procedure that can increase the allowed number of errors for a specific column, given the number of errors corrected so far. This was particularly important, as the authors note that HapCol could fail to find a solution if multiple sites had a true error rate above k (*i.e.* α was higher than estimated).

Additionally, like WhatsHap, HapChat's pre-processing step artificially reduces the coverage by merging reads that are assumed to originate from the same haplotype, assuming uniformly distributed errors at a fixed rate of $p \approx 0.15$. For every pair of reads r_u and r_v, the likelihoods that the reads originate from the same haplotype (or not) are calculated. Read pairs are then clustered by the ratio of these likelihoods, and *sets* of reads are only collapsed with per-position majority rule if the cluster meets a threshold likelihood ratio. The authors note that this method is more conservative than ProbHap, which only considers when to merge two reads, rather than sets of reads, improving the quality of the merged data.

HapChat's novelty appears to be limited to the combination of WhatsHap's read pruning, and a recurrence relation applied to HapCol's per-column error correction minimisation. Despite this, the results show that the approach appears to overcome the limitations that hampered both of its

[65] A monomial matrix has one non-zero entry in each row. A monomial submatrix in this context shows the two fragments r_u and r_v have an incompatible phasing over sites i and j.

predecessors, outperforming not only `WhatsHap` and `HapCol`, but also achieving switch error rates and Hamming distances[66] as good as, or better than the recently released `HapCUT2`. The authors note that it was not possible to compare `HapCUT2` on real data, as a preprocessing realignment step could not be completed, although it is unclear as to why this is the case[67]. The work awaits peer-review.

The summary of more recent approaches from the past year or two make it clear that **sequencing depth is still one of the most fundamental requirements for accurate single individual haplotyping**, and tools that can scale to handle more information at each SNP position are more capable of accurate haplotype recovery. `ProbHap`, `WhatsHap` and now `HapChat` *imitate* the support for coverage with a lossy procedure that considers whether read fragments should be merged first. Future algorithms that can actually handle coverage without such preprocessing will likely yield better haplotype recoveries.

2.5.9 Discussion

We are at an interesting time for single individual haplotype recovery, with modern long read sequencing technologies renewing interest in SIH for the reconstruction of whole-genome haplotypes. We have seen that statistical methods have greater versatility (such as swappable models for technology-specific read errors) and accuracy (finding the maximum likelihood haplotype appears to be more accurate than optimising flipped elements in M with MEC). Though, it is clear that the literature is diverging, with new tools moving towards more bespoke SIH formulations that consider the idiosyncrasies of the sequencing approach used to generate the reads themselves. Some surveyed approaches are demonstrated to be exponential in resource requirements as coverage increases (as read length is assumed to grow faster), allowing some previously intractable approaches to fall back in fashion, given the low depth of long read data.

We have almost come full-circle since the anecdote found in the discussion of Levy *et al.*'s work on *HuRef* (Section 2.3.2) – where the authors noted the difficulty was not so much in the assembly of the haplotypes, but in the sequencing protocols that allowed those haplotypes to stay intact in the first place [159]. One of the goals of modern long read technologies are to make the assembly problem easier, with fewer, longer fragments to piece together. Current error rates demonstrate there are still clear difficulties in keeping these long molecules in one piece, with integrity, such that they can be sequenced with accuracy [195, 196]. Current approaches appear to be geared towards identifying and resolving these specific errors.

Of course, as described by Ahn and Vikalo in their introduction of `ParticleHap`, many of the surveyed approaches still formulate the problem as one that considers no more than two alleles at each SNP position, discarding tri- and tetra- allelic sites, reducing the accuracy of the recovered haplotypes.

[66] Briefly; Hamming distance describes the between a pair of strings. Refer to 4.1.2.

[67] I would hope any future accepted version of this manuscript would have attempted to contact Edge *et al.* to determine whether this is a bug to enable a proper comparison.

All the probabilistic methods introduced here; from Li *et al.*, `MixSIH` (Section 2.5.4), `ProbHap` (Section 2.5.5) and `ParticleHap` all feature component distributions that collapse to $\frac{1}{2}$. `HapCUT2`'s formulation of read fragment likelihoods divides the probability of a fragment element featuring in either of the haplotype's by 2. Indeed, even if one could assume a different ploidy, and model that distribution in place of these constants, the algorithms still work to recover just one (or sometimes a pair) of haplotypes, typically with opposing heterozygous alleles. More so, the fundamental approach of the algorithms too assumes a solution in this format; `HapCUT2` considers the improvement of a cut that defines element flips between the haplotype pair, `HapChat` must find the minimum flips to make in a column such that column pairs are not in conflict, `ProbHap`'s formulation can only recover one binary haplotype, and both `ProbHap` and `MixSIH` scale poorly with the coverage that would be necessary to recover multiple haplotypes.

Ultimately, **these tools continue to build on the underlying problem first introduced by Lancia in 2001 (Section 2.2): that read fragments, or SNPs, or elements of both are in conflict with one-another, preventing a solution of just two haplotypes**. Thus no such work presented so far is capable of addressing the problem of haplotype recovery in the context of polyploids or metagenomes.

2.6 Toward polyploid-capable methods

As discussed at the end of my previous section, the methods introduced so far consider diploid organisms only. Kim *et al.*'s 2007 work (Section 2.5.3) did consider a future whereby their haplotype recovery algorithm may wish to target polyploid assemblies, noting it as more complex, even in the error free case. Indeed, considering polyploids adds difficulty from the outset, as a read fragment favouring one phasing solution can no longer be assumed to also offer evidence in favour of the complementing phasing, or contradict other alternate phasings. I briefly present more recent work that moves towards support for polyploid data.

2.6.1 HapCompass with polyploid support (2013)

In 2013, Aguiar and Istrail were **the first to implement an efficient heuristic to address the problem of polyploid reconstruction** by extending their previously described work: `HapCompass` (Section 2.4.2) [197]. The literature of polyploid haplotype assembly is described in their paper as "essentially non-existent" and the authors argue that we have been hindered by the basic assumption found in every tool surveyed that only two phases between a pair of SNPs can exist. The authors additionally note that polyploid assembly is similar to that of 'quasispecies identification' and 'metagenomics modelling'; both problems that I will come to discuss later. While the manuscript goes on to prove some interesting properties about their previously introduced structure and MWER optimisation

(minimum weighted edge removal), including the inevitable proof of its NP-hardness, I consider only the section of their method which extends the structure to consider polyploids.

The formulation of the SNP matrix M still features binary elements (and the gap $-$), from which Aguiar and Istrail reintroduce their compass graph G_C; a structure in which each SNP is a vertex, and pairs of SNP vertices are joined by an edge if at least one read fragment overlaps them. `HapCompass`' underlying axiom was every edge between a pair of SNPs had a unique phasing that could be computed from the reads observed in M.

The compass graph is constructed from M as before. For each heterozygous SNP site, `HapCompass` appears to estimate the relative quantities of alleles for each of the k haplotypes. **The number of haplotypes must be known by the user in advance**. Given the inferred genotypes for a pair of positions, `HapCompass` calculate the possible haplotype phasings. Describing their own example, consider a tetraploid, and a pair of positions with genotype $\{0,0,1,1\}$ (*i.e.* a relative abundance of 50% for both the 0 and 1 allele in the corresponding columns of M), one can enumerate the three potential phasings of the four haplotypes[68]: $\{(00,00,11,11),\ (01,01,10,10),\ (00,01,10,11)\}$. Each of these three possible phasing configurations contain 4 *haplotype bins*.

The haplotype bins represent the k phased haplotypes, for each of the possible phasing configurations between a given pair of SNPs. `HapCompass` offers both a greedy and probabilistic binning approach for populating the bins with read fragments to select the configuration with most support, though later results show the probabilistic method yields better results, so I describe the latter only. Given two SNPs i and j, we denote the set of possible phasing configurations as $P_{i,j}$, `HapCompass` computes the conditional probability of each $p_k \in P_{i,j}$ being observed, given the read fragments ($r \in R$) and a fixed sequencing error rate (ε). This is non-trivial: each p_k requires the calculation of the conditional $\mathbb{P}(p_k \mid \varepsilon, r_1, ..., r_m)$; which considers both the product (and the the sum of products to divide by) $(\sum_{p_k \in P_{ij}}) \prod_{i=1}^{m} \mathbb{P}(r_i \mid \varepsilon, p_k)$, and each $\mathbb{P}(r_i \mid \varepsilon, p_k) = \sum_{b_l \in p_k} \mathbb{P}(b_l \mid \varepsilon, p_k) \times \mathbb{P}(r_i \mid \varepsilon, b_l, p_k)$. The maximum likelihood $p_k \in P_{i,j}$ configuration is selected to phase the pair of sites.

For the consideration of polyploids, `HapCompass` also introduces the *chain graph* G_h. G_h is constructed by considering a path or cycle of length l through the compass graph G_C. Consider a length l path over a set of edges in the compass graph G_C. Denote each edge $e_i \in \{e_1..e_l\}$ along the path, and consider it is phased with some $p \in P_{i,i+1}$, with k haplotype bins $b_l \in \{b_1..b_k\}$ in the phasing. Each such bin adds a vertex of the chain graph, and each edge adds a 'level' i. Thus the chain graph has a total of $k \times l$ vertices.

A pair of haplotype bin vertices are joined in the chain graph if they both overlap a SNP position, and allele. Recall an edge defines the phasing between a pair of SNP sites, and that each haplotype bin in an edge phasing defines the allele across the two positions, so by definition, edges in the chain graph can only exist between adjacent levels. Thus, any path through a chain graph corresponds to an

[68] Note that $\{00,00,11,11\}$ is equivalent to $\{11,11,00,00\}$ or $\{00,11,00,11\}$ *etc.* as the haplotypes themselves are unordered. This is not made clear in the manuscript and led to some frustrating scribbling of many examples.

extension of haplotypes across multiple SNP pairs. Of course, the previous version of `HapCompass` was concerned with finding cycles through the compass graph (Section 2.4.2). Here, cycles are represented with k 'source' and 'sink' nodes in the chain graph. Each source node is arbitrarily connected to one of the k haplotype bins on the first level. However, sink node i is only connected to haplotype bins on level l if those bin can be joined to the corresponding source node i. Thus, the problem of polyploid haplotyping can be reduced to the problem of determining k disjoint paths[69] through the chain graph.

A depth first search (DFS) is used to find the k disjoint paths through the chain graph, by constructing an additional auxiliary *flow graph* to summarise the possible paths between sources and their corresponding sinks. Conflicting cycles can be identified if there is no such path between a source and sink node. I leave further specifics of this to the original manuscript, but the maximum flow through this graph corresponds to the valid assignment of the k haplotypes.

The polyploid evaluation is conducted with simulated data, from just three related haplotypes. The results show good accuracy, but it is measured by the number of correctly phased SNP pairs, rather than a quality metric such as Hamming distance to the actual haplotypes, so it is unclear whether the extended phasings are recovered in the correct order. Accuracy is strongly correlated with the availability of sequence coverage, ranging from 80% to ≈95% as the number of reads varies from 100 to 10,000. `HapCompass` **offered the first proper formulation of SIH for polyploids, but the algorithm still assumes SNP sites are biallelic**; precluding its usefulness to mixed samples such as metagenomes, or highly variable viral genomes. Additionally, as described, the binning step requires a considerable number of calculations, dependent on the number of haplotypes and possible phasing configurations for each SNP pair – which too increases with the number of haplotypes. Indeed, the manuscript notes that when extending phases between adjacent edges in the compass graph, alleles that are present in k haplotypes will yield $k!$ valid potential extensions, making it unsuitable for situations where there are many haplotypes to recover.

2.6.2 HapTree (2014)

A year after `HapCompass`, Berger *et al.* also identified that the problem of polyploid haplotyping had received "little attention", and introduced their own maximum likelihood method: **HapTree** [198].

Berger *et al.* recognise `HapCompass` as the current 'pioneering' state-of-the-art for polyploid reconstruction, but notes that it phases SNP pairs first, and extends them with the chain graph afterwards. Unlike `HapCompass`, `HapTree` aims to perform pairwise SNP phasing during the assembly of the haplotypes themselves. Like previous work, the approach still assumes that each SNP position is biallelic, and additionally requires that the genotype for each SNP is available. The authors formulate

[69] A set of k paths are vertex-disjoint (or vertex-independent) if they do not share any vertex in common. Here, the k paths must traverse through the k haplotype bins at each level of the chain graph, without both selecting the same bin (though, multiple bins at the same level can have the same value).

the problem as that of reconstructing the most likely phases given the SNP matrix M, the known genotypes of each SNP position, and knowledge of sequence error rates.

`HapTree`'s approach is to find the maximum likelihood phasing of the first j SNPs, and extend this phasing to $j+1$ SNPs, iteratively reconstructing the haplotypes simultaneously in this fashion, repeating a branching and pruning step to consider each additional column in M. Interestingly, `HapTree` considers read evidence only up to the current position to which it is trying to extend. The function $SR(R,i)$ constructs the set of partial read fragments $r \in R$ ('semi-reads') that cover positions $\leq i$, truncating any parts of reads that extend beyond i.

`HapTree` begins by initialising the set of current haplotypes $H = \emptyset$ and the first branch step extends this empty set onto the first SNP by selecting a random permutation of the first genotype. Future branching considers the collection of extensions between a current position j and $j+1$, that could be applied to each $h \in H$. Each allele in the genotype of SNP $j+1$ is appended to each haplotype in H in turn, and `HapTree` considers whether the resulting $j+1$ length haplotype meets a likelihood threshold via $\mathbb{P}(h_{j+1} \mid h_j,\ SR(R, j+1),\ \varepsilon)$. This is the conditional probability that the new haplotype h_{j+1} is correct, give the assumption that h_j was correct, the set of semi-reads that cover $j+1$ (but no position beyond) and a fixed sequence error rate (ε).

The likelihood of a phase v ("vector set") is defined similarly to that found in the approach of `HapCompass`, again requiring calculation of the conditional $\mathbb{P}(v \mid \varepsilon, r_1, ..., r_m)$, and the similarly defined product over all reads. The conditionals for each read $\mathbb{P}(r_i \mid v, \varepsilon)$ in this instance construct a probability by considering the number of positions where the read fragment and phasing vector agree or disagree, with the context of potential fixed sequencing errors at each position. Any haplotype extension that meets a user-defined threshold is added to the set of haplotypes, which is then pruned in the following step. Pruning considers the highest likelihood haplotype currently in H, and removes all haplotype candidates whose own likelihoods do not fall within some distance to the current best. The specific considerations of both the extended haplotype and pruning acceptance conditional probabilities are not explained sufficiently in the manuscript, and are left by the authors to be derived from the phase likelihood probabilities described. Note that the haplotype extension and pruning thresholds must be provided by the user.

During evaluation, Berger *et al.* highlight **both MEC and switch error are inappropriate measures of accuracy for polyploid recovery**. MEC effectively measures the distance between the corrected matrix M and recovered haplotypes; but as described earlier by Aguiar and Istrail and remarked here by Berger et al. the apparently phase of an observed read fragment cannot offer evidence for or against another phasing in the case of non-diploids, making it hard for such a distance metric to consider whether an element conflicts with the haplotype configuration. Additionally, switch error will overly penalise error on subsets of the haplotypes, as a switch between a pair of haplotype vectors has the same weight as a switch between multiple pairs of vectors.

As an alternative, Berger *et al.* offer the definition of *vector error* instead. Vector error considers the minimum number of switches that must occur across all haplotypes. As a generalisation of switch error, one can consider the vector error of a diploid recovery as twice the measured switch error.

Berger *et al.* compare `HapTree` to `HapCompass`: "the only other existing program that directly addresses polyploid assembly". Evaluation is performed on simulated data[70] at various factors of genome size, and sequencing coverage. The authors simulated triploids and tetraploids.

Regardless of coverage or number of SNP loci, the presented figures indicate a significant improvement over `HapCompass`, with the smallest improvement between the two programs measured at 63% on the simulated triploid. With the tetraploid, Berger *et al.* found that `HapCompass` recovered haplotypes with a pairwise SNP accuracy of less than 1%. Although improved, `HapTree`'s own performance is between 10% and 65% as coverage increases from 10× to 40×. These results appear in contrary to those presented in the `HapCompass` manuscript, owing perhaps to differences in the method of simulation in both papers. Berger *et al.* argue that **considering likelihood scores rather than optimisation of MEC yields improved haplotypes and the MEC is not suitable for optimisation of polyploid solutions**.

Their figures show "perfect reconstruction rate"[71] and vector error degrades as ploidy increases beyond $k = 2$. `HapTree` relies significantly on good read coverage, with 40× coverage required to achieve SNP pair accuracy scores of around 85% on the triploid, and 65% on tetraploids. The **results consider a mere 10 SNP sites in simulated blocks, and the largest blocks contained just 40 SNP loci, making it difficult to see how such an approach could scale**. Despite the authors comment that their tool could also achieve better phasings on diploid problems than `HapCUT`, it was shown during the evaluation of `HapCUT2` (Section 2.5.7) to not be effective enough to consider in their comparison.

The Discussion adds that the approach could be "easily extended" to handle multi-allelic positions, and cases where the genotype is not known for each SNP in advance[72]. However, I would question whether the formulation provided could scale efficiently; with 4^k possible allele permutations distributed over k chromosomes if one permits the SNP matrix to hold 4 alleles, rather than 0 or 1. Without handling **non-binary elements in M; `HapTree`, like `HapCompass` is not suitable for recovering haplotypes from a mixed sample, like metagenomic data**.

[70] With Berger *et al.* remarking "real polyploid data is hard to come by". The authors echo the problems observed over a decade ago by Panconesi and Sozio (Section 2.3.1) and others who remarked that the formulation, development and testing of algorithms for single individual haplotyping was difficult with a lack of publicly available sequencing data.

[71] Which is merely a nice new name for the pairwise SNP reconstruction rate reported in the `HapCompass` manuscript.

[72] Yet, it has not been done.

2.6.3 Lens (2016)

In 2016, **Lens**, a new haplotyping approach was published, as part of a landmark 2016 *Nature Biotechnology* paper that demonstrated diversity within bacterial strains in a human gut microbiome with the use of "synthetic long reads" [199]. The lead author of the work was Kuleshov, who also published the previously surveyed `ProbHap` (Section 2.5.5); evidently continuing along the theme of SIH algorithms targeted at long-read data. The approach itself is detailed in the article's monolithic supplement [199].

`Lens` is **designed to operate without assumptions on the length of input reads, or ploidy of the underlying sample**, identifying that with few exceptions (one being `EVORhA`, Section 2.7.2) tools that are capable of handling polyploid data (such as the updated `HapCompass`, Section 2.4.2) require the ploidy to be known in advance, which had hampered our ability to recover bacterial haplotypes from a microbiome.

With the use of long-read technologies (synthetic or otherwise), `Lens` benefits from the fact that a single long read must have originated from one particular organism. The greedy assumption is that reads which share the same mutations must therefore also originate from the same organism, or strain. By connecting these long reads at their overlapping sites, multi-kilobase haplotype reconstructions can be generated.

The approach is simple, and greedy, and reminiscent to the algorithm of `FastHare`[73] (Section 2.3.1). The first step considers variant calling: given a set of co-ordinate sorted reads, aligned against some reference, `Lens` first calls for variants between the reads and the reference; requiring any variant position to be supported by at least 3 reads, and a 5% minimum minor allele frequency. These positions are important, as it is these variants that are used to determine closely related bacterial genomes for reconstruction later. Although the manuscript does not do so, one can formulate this structure as Lancia's SNP matrix M (or rather, Li *et al.*'s version), by considering each long read fragment as a row, and each of the called variant positions as a column. Elements $M[i][j]$ will then contain the nucleotide (or gap) observed at position j of read i.

Haplotypes are reconstructed by considering each of the reads that overlap a variant position, from the start of the reference to the end, in turn. The first read observed at the first variant position forms its own haplotype, then for each variant position s_j ($j = 1..n$), given each of the read fragments r_i that span j, `Lens` decides whether to use r_i to extend a current haplotype, or to use the read to form a new one. If a read overlaps one of the current haplotypes a user-specified number of different SNPs ($\leq j$) without error, then the read is used to extend that particular haplotype, otherwise the read is added to the set of current haplotypes and is considered for extension by future reads. Once all reads have been processed, the collected haplotypes are processed to observe whether they meet user-specified criteria on the breadth and depth of read coverage. `Lens`' default parameters require strict matching,

[73] As the saying goes, "everything that is old is new again"

and will extend a haplotype if the read overlaps at least 2 SNP positions, and has 100% similarity at all overlapping positions[74].

It is of note that `Lens` **performs its own variant calling**, echoing the idea behind `ParticleHap`: that more accurate haplotypes could be generated by ignoring the genotypes produced by current tools. Of course, given that Kuleshov *et al.* were trying to recover haplotypes in a non-diploid setting it is perhaps more likely that they recognised that no current SNP caller would yield satisfactory results (as observed by Ahn and Vikalo). Given the strict matching requirements, `Lens` arguably relies on high quality, high coverage data, which was obtained in this study with application of the Illumina Tru-Seq protocol; extracting kilobase-long DNA fragments for amplification and reassembly with short reads. Interestingly, the work also performed regular short read Illumina sequencing in parallel, and discussed the differences observed across the two technologies, and suggested future works should consider using both technologies to complement one another.

Clearly, relaxing the matching parameters for a run of `Lens` will increase the number of returned haplotypes. Kuleshov *et al.* show that `Lens` returns thousands of haplotypes for their data set and a detailed discussion on their length and overall quality is missing (although admittedly, there are many results to fit in the manuscript). It is however mentioned that recovered regions contained an average of 3.93 haplotypes, and I would wonder to what extent the post-processing haplotype pruning parameters remove subtle variants.

`Lens` showed that future work must move away from the assumptions made by the haplotype assembly tools of the time, and clearly demonstrates their approach is applicable to metagenomic data. However, **its greedy bias generates many unordered haplotypes and provides no mechanism for a user to conduct ranking or selection**.

2.6.4 Discussion

`Lens` has not been further developed since its initial release in 2016, and still fails to scale well with coverage, or provide a method for selecting likely haplotypes. Although the new release of `HapCompass` is capable of supporting the recovery of polyploids, note that it requires the ploidy to be known, which is not a suitable or answerable question for metagenomic data. `HapCompass` and `HapTree` still formulate their problem with binary alleles, forcing the non-reference allele to be the reference complement, reducing its suitability for metagenomic data.

Ultimately, these tools strive to answer a different problem to that of investigating a microbiome. Given the evaluation of `HapTree` (and its comparison to `HapCompass`), shows accuracy quickly reduces as more haplotypes must be recovered, and scales to handle a small number of SNPs per haplotype, one could argue the research into polyploid recovery is still in its infancy, but holds promise for our insights into complex plant and animal genomes in future.

[74] https://github.com/kuleshov/lens/blob/master/detect_subspecies.py#L38

2.7 Viral quasispecies reconstruction (VQSR)

In parallel to the surveyed work on the single individual haplotyping (SIH) problem, the related problem of *viral quasi-species reconstruction* (**VQSR**[75]) has become a hot topic [102, 200]. An infection of viruses such as HIV, Zika and Ebola populate their host with an ensemble of highly related mutant strains, which can be considered as haplotypes. Reconstructing the sequence differences among a population of mutating viruses presents challenges similar to the SIH scenario. Akin to the problem of reconstructing a metagenome, the true number of strains is effectively unknown.

I present an overview of some of the recent advances in this related area.

2.7.1 ShoRAH (2011)

Recognising the benefits of deep sequencing offered by modern sequencing technologies for the problem of investigating the populations of mixed samples, Zagordi *et al.* presented their computational approach designed to identify and correct errors, and estimate haplotype frequencies with their novel toolkit: *Short Read Assembly into Haplotypes* (**ShoRAH**) [201]. `ShoRAH` appears to be a rebranded version of Zagordi *et al.*'s work from a year earlier in 2010 [202]: pre-dating the majority of previously surveyed work.

This prior work introduced a probabilistic model for assigning read fragments to haplotypes, with consideration of sequencing errors. Zagordi *et al.* imagine sequencing as a statistical process, where reads are drawn from an underlying population of haplotypes. Modelling this distribution allows one to make inferences about the originating sample's haplotypes. The goal of `ShoRAH` is to slide a moving window over a set of aligned read fragments, using the observed read evidence to infer local haplotypes, and use the local reconstructions to describe haplotype frequencies for the whole population. Like the majority of the probabilistic formulations surveyed in this review (Section 2.5), the haplotypes (h) and assignment of read fragments to haplotypes (a) and some form of sequencing error must be modelled. Unlike the previously described work (with the exception of work presented in Section 2.6) `ShoRAH` was designed to be capable of considering more than just possible haplotypes.

The problem is formulated as one of clustering reads in a window together by their similarity, attempting to separate noise introduced by sequencing error away from true biological variation. `ShoRAH` requires a multiple sequence alignment, with reads considered against a common reference genome as well as each-other. This is a computationally expensive process, but the toolkit is designed to target specific regions of a genome, rather than the entire reference sequence.

A mixture model (similar to that later used by `MixSIH`; Section 2.5.4) is used to estimate the prior distribution on the number of unknown haplotypes in the sample. The model controls the number of classes (haplotypes) that are created when attempting to cluster the observed read fragments in a given

[75] Not to be confused with Variant Quality Score Recalibration

window. Using the mixture model, ShoRAH can measure $\mathbb{P}(R \mid a,\ h)$: the probability of observing the read fragments[76] $r \in R$, given the haplotypes h and a read assignment vector a. This distribution is similar to the sequence accuracy component of many of the surveyed approaches (described first by Li *et al.*). Of course, this distribution must consider the product *over all haplotypes*, where each product considers the sum of probabilities that one may observe a read fragment given it was assigned to that haplotype.

For local analysis, each window (which is set to be the size of an average sequenced read) over the multiple sequence alignment is processed by a Gibbs sampler (very similar to the formulation previously introduced by Kim *et al*'s prior work in 2007; Section 2.5.3). The sampler allows ShoRAH to draw from the unknown joint posterior distribution that considers the haplotypes, read assignments and sequence error; alternating estimations of each of these components separately to converge on a high likelihood estimate of each probability distribution. In brief, for each run of the sampler; reads are assigned to existing haplotypes (or create new ones according to the prior), haplotypes are then sampled given the read assignments and sequence error, finally sequence error is estimated given the new read assignments and haplotypes. Once the sampler has iterated T times (where an appropriate T must be chosen by the user), the original fragments are corrected according to the majority rule of the collection of reads in the same cluster (haplotype).

The corrected read fragments are then processed by ShoRAH's global reconstruction step. Little information on this step is found in both manuscripts that describe the approach, but it appears to attempt to find the maximum substring shared across the reads for each of the clustered haplotypes. Finally, the frequencies of each of the haplotypes are estimated to describe the population structure.

Despite promising results for locally reconstructed (*i.e.* read length) haplotypes, **the evaluation of the global reconstructions consists of one paragraph**, describing that the quality of recoveries from error-free simulated reads depended on the number of present haplotypes, the sequence similarity of the haplotypes. No actual results for global reconstructions are presented in either manuscript. Additionally, the authors note that for short Illumina reads, the global reconstruction step can be unstable. However, the work shows that ShoRAH is capable of predicting the number of haplotypes and performing error correction on sequencing reads with reasonable accuracy; **without prior knowledge on the number of haplotypes or the error rate**.

ShoRAH offered a first formulation of probabilistic quasispecies reconstruction (echoing similar concepts to previously described work) that could describe the number of haplotypes and their frequency in an underlying sample. Later reviews on early approaches to VQSR [203, 204] found **ShoRAH's results to be sensitive to its given parameters, and that it typically returned too many haplotypes ("greatly overestimating the population size") with low accuracy**. The review additionally confirmed that short read data sets were unsuitable for use with ShoRAH and that in many practical cases, they could not run the algorithm.

[76] Arguably, this could still be considered as a SNP matrix M

2.7.2 EVORhA (2015)

In 2015, Pulido-Tamayo *et al.* recognised that in non-viral clonal systems, fewer SNPs adds difficulty to the problem of recovering haplotypes, and **claim that no current technique can recover haplotypes from a bacterial population**. To overcome this, they introduced *Evolutionary Reconstruction of Haplotypes* (**EVORhA**) [205], using frequency ratios to complement local phasing information to construct genome-wide haplotypes. Although not specifically designed for the recovery of viral haplotypes, the work presents itself in the context of VQSR, and additionally compares itself to state-of-the-art viral quasispecies recovery approaches, and so fits in here with the overall narrative.

Inspired by `ShoRAH`, `EVORhA` combines local haplotype inference with error correction with a novel probabilistic approach for genome-wide reconstruction that takes frequency information into account. Using frequency information for local haplotype extension permits `EVORhA` to be used in cases where pooled clonal samples have a low mutation rate, yielding reads with little-to-none overlapping evidence, or for the use of short-read based sequencing technologies (where `ShoRAH` is unsuitable [203, 204]).

Given a set of sequenced reads aligned to some common reference[77], `EVORhA` begins by identifying windows along the alignment that meet some user-specified criteria (window size and read depth; defaulting to 60% of an average read and 30× respectively). For each window w, the set of possible 'template' haplotypes h_w is generated from the reads that span the entire window; where any novel combination of SNPs along full-fragments in the window can be considered for h_w.

The level of fragment support for each template haplotype $h \in h_w$ is then found by counting the number of full-length reads within a window that support each haplotype. Partial reads are allowed to contribute lesser-weighted secondary support to any haplotype they are consistent with. All primary and secondary haplotype supports are weighted by the lowest quality position of the read fragment. An oddly specific equation defines the support threshold that each haplotype must obtain to not be pruned from h_w. The threshold increases with support, and also considers the 'severity of amino acid changes' that polymorphisms on the haplotype will cause. Potential haplotypes with higher coverage in the window, or less likely amino acid changes therefore need more support than others. Error correction is conducted as part of thresholding by reassigning read fragments that contributed to pruned haplotypes to the most evolutionary related template haplotype that has been accepted.

Window extension begins with a 'seed', the window with the highest number of accepted haplotypes, polymorphic sites and coverage. `EVORhA` then concatenates haplotypes across its flanking windows. For a pair of overlapping windows w_a and w_b, the set of potential partial intersecting haplotypes enumerating the unique combinations of polymorphisms observed in the overlap of *both* windows is constructed. Each window's accepted template haplotypes are assigned to one such partial overlap. Where a partial overlap does not receive an accepted haplotype from both windows (to extend), the

[77] Which again, can be formulated as a SNP matrix M.

orphan is partnered with the closest evolutionary haplotype (by sequence similarity). If a partial overlap is assigned multiple haplotypes from both windows, the extension is ambiguous and an expensive iterative heuristic step that considers all possible assignments is conducted to maximise the likelihood of the haplotype extensions[78]. This continues until all windows have been extended as far as possible.

The previous extension method is similar to that found in ShoRAH. However, unlike its surveyed alternatives, EVORhA now attempts full genome-wide haplotype reconstruction, with the goal of bridging any remaining gaps across extended windows where no further read evidence can be used to join haplotypes together. EVORhA considers the frequencies (*i.e.* read supports) of extended haplotypes across adjacent blocks, and attempts to group them together by considering a Gaussian mixture model[79], where the underlying assumption is that the read sets that support each haplotype in a maximally extended window can be represented by k Normal distributions, each of which are assumed to be stable over multiple extended windows and are thus unique. Starting from an initial seed, the method effectively assigns each extended haplotype to a Normal (Gaussian) distribution if its observation frequency falls within one standard deviation of the mean of one of the existing distributions, or creates a new distribution for the mixture.

However, the true haplotypes could still be described by the mixture. A post-processing method considers each pair of extended haplotypes that have been assigned a single distribution and calculates a distance based on the difference between the size of the union and intersection of their polymorphisms. The final inference is a two-step process considering whether the pairwise means and corresponding distances indicate that two distributions should be considered together, and finally the tool jointly reports the haplotypes whose frequency sum falls within a 95% confidence interval for the same distribution, yielding the maximum likelihood haplotype extensions, and thus the haplotypes.

Performance was assessed with simulated data, with an average reliability score ranging between $70-80\%$ as coverage increases[80] from $50\times$ to $500\times$. Additionally, as the inspiration for EVORhA was the problem of VQSR, the authors chose to compare the method to ShoRAH on simulated viral genomes. With longer reads ($\approx$ 700 bp), the state-of-the-art tools ShoRAH (Section 2.7.1) and QuasiRecomb marginally outperform EVORhA. With short Illumina reads ($\approx$ 100 bp), EVORhA offers a significant improvement in accuracy ($\approx 30\%$). However, the simulations only consider haplotypes featuring up to 50 SNPs. Additionally, despite pre-existing early literature on polyploid recovery (Section 2.6), EVORhA is not compared to the surveyed methods.

Although EVORhA **showed that current state-of-the-art VQSR methods are not suitable for larger bacterial genomes**, its second step **"assumes that the haplotypes in the population have developed from a common ancestor"**. This allows the second step to generate extended haplotypes based

[78] This step maximises a Poisson distribution to model the emission of read fragments in favour of an extended haplotype, from the accepted haplotypes across both windows

[79] See Section 2.5.3 or 2.7.1 for other similar applications of mixture models.

[80] Amusingly, $50\times$ is considered "low coverage", whereas the counterpart problems in SIH would consider this sufficient.

on frequency information, but **the phylogeny of haplotypes for a region of a metagenome will not necessarily all share a common ancestor**, thus `EVORhA` can fail to recover true haplotypes in a non-clonal setting where this assumption cannot be assumed true, such as a metagenomic sample. Additionally, reads assigned to a poorly supported haplotype, or partial haplotypes between two windows that are orphaned, are both reassigned based on phylogeny, which too can be undesirable in a non-clonal setting. Indeed, the Discussion highlights that **the approach could be appropriate for metagenomic data but only if one first reduces complexity with binning approaches**. The number of haplotypes that can be returned is also directly affected by the coverage of the sample, as a low coverage leads to large standard deviations in the Normal distributions that make up the Gaussian mixture model, which can lead to distinct haplotypes being incorrectly considered together as a result of the final step. Of course, any haplotypes that occur at similar frequency pose a problem for the algorithm.

2.7.3 SAVAGE (2017)

Citing the high mutation rates and observed diversity within viral samples, Baaijens *et al.* note that available reference sequences can be obsolete when attempting to align reads from an individual infection and remark that this lack of a suitable reference sequences continues to pose a major hindrance to many VQSR approaches [206]. To circumvent the requirement for a high-quality reference, Baaijens *et al.* introduced *Strain Aware Viral Genome Assembly* (**SAVAGE**): **a *de novo* assembler for viral genomes** (first shown possible by a prior preprint `MLEHaplo` [207]). As generic assembly approaches attempt to generate consensus sequences from the input reads, `SAVAGE` instead uses the higher rate of polymorphisms in its favour to directly assemble haplotypes from viral samples.

The introduction of the work describes the failing of generic *de novo* assemblers for haplotyping falls onto the use of the *de Bruijn* graph representation (Section A.4.1), arguing that detection of co-occurring patterns is improved when one can consider reads in their full length, rather than the k-sized substrings that sequence reads are broken into for *de Bruijn* assembly. For this reason, `SAVAGE` uses *overlap graph* assembly[81] (Section A.4.1), which was first used for viral assembly in another prior work (`HaploClique` [208]) but was found by Baaijens *et al.* to function poorly as its complexity was exponential as coverage increased[82].

The algorithm proceeds in three parts; in a similar fashion to previously described works: local, global and master haplotype assembly. Each step requires the construction of an overlap graph. Local assembly uses the sequenced reads, and both global and master assembly considers the overlapping contigs derived from the previous step. The overlap graph G_O contains a vertex for each read (or contig). A pair of reads r_i and r_j (or contigs) are joined by an edge if a function $QS(r_i, r_j)$ determines the suffix of r_i overlaps the prefix of r_j has a quality score that meets the threshold δ, and if the length

[81] Again, "everything that is old is new again".
[82] Describing 1000× coverage as "relatively low"

the overlap is at least l. The edge is directed from r_i towards r_j. *QS* borrows the statistical model presented in the `HaploClique` manuscript, to calculate the likelihood that the read (or contig) pair arise from the same haplotype[83]. The parameters for thresholding (δ), minimum overlap length (l) and permitted mismatch rate (defaults to 0, though master-mode permits divergence of 1%) must be set by the user.

The problem at hand is that **overlap graph approaches require the expensive computation of all pairwise read overlaps** (Section A.4.1), which was in part the reason that focus shifted towards the more efficient *de Bruijn* graph representation as modern sequencing experiments began to sequence with deeper coverage. For the global and master steps, the number of overlaps to perform is usually small and manageable, as `SAVAGE` strongly constrains overlaps to reduce error and produce conservative overlaps for more efficient computation. However, the local step must find the overlaps of millions of deep-sequencing read pairs. `SAVAGE` offers two options to compute the overlaps: the first determines all overlaps that meet the criteria from an *FM-index*[84], and the second considers the alignment of reads against a reference genome.

Given an overlap graph, the core idea of the approach is to compute *cliques*[85] (again inspired by the pre-existing work `HaploClique`). In the first instance, `SAVAGE` must prune edges for G_O to simplify its structure, as the number of maximal cliques grows exponentially with the number of fragment vertices. After this, reads are clustered together by finding cliques of maximum size in the pruned graph, and are "carefully" flattened into consensus contigs: like many methods, a position-based quality-weighted majority vote is conducted. This generates the partial haplotype from the clique, and corrects suspected errors.

`SAVAGE`'s assembly approach considers repeated application of the clique-finding and clique-merging steps, with the new consensus contigs forming the input for the next overlap graph, until the overlap graph contains no edges (*i.e.* cannot be assembled further as no overlaps exist).

`SAVAGE` was evaluated against `ShoRAH` (Section 2.7.1) and `PredictHaplo`. The authors could not assess performance against `HaploClique`, as it was no longer supported. The manuscript presents many tables and figures to describe its evaluation over six data sets. The performance of `SAVAGE` against its competition on the five simulated data sets is undoubted, constructing contigs that cover over 99% of the target genomes, with the fewest mismatches and indels of the four algorithms.

[83] The method is simple and effectively calculates the likelihood of the overlap of the two sequences, weighting the probabilities with their associated base quality scores (in a similar fashion to many of previously surveyed works), but does also consider the probability of two hypothetical reads or contigs supporting the provided sequence pair over the positions where they do not overlap (using the marginals to model the probability of observing a new pair of sequences that would agree with the non-overlapping parts of the two provided sequences) [208]

[84] *Full-text indexing in minute space* allows dense compression of input data, while retaining the ability to quickly query for the count and positions of a given substring.

[85] Subsets of vertices in G_O whose pairs are adjacent (*i.e.* each pair of the subset is connected by an edge)

Importantly for the context of this thesis, their evaluation shows that **general-purpose assemblers struggle to identify haplotypes, including** `metaSPAdes` **– the state-of-the-art metagenome-aware assembler**, backing up the argument **that more specialised assemblers are necessary for strain-aware assembly**.

However, **when considering a real laboratory mix consisting of five known HIV haplotypes,** `SAVAGE` **constructs hundreds of contigs (an order of magnitude more than its competition)**, with an N50 of only 500 bp, in a scenario where only five haplotypes exist. Indeed, the manuscript refers to this as "the most challenging benchmark", but the discourse on the result consists of two paragraphs and makes no comment on their own performance. Whilst the authors note that the number of contigs returned by `SAVAGE` (and the other *de novo* assemblers) cannot be interpreted as the number of strains predicted to be in the sample, one must note that hundreds of contigs (with length $\geq$ 500 bp) for a sample that should contain only five haplotypes with a length of $\approx$ 10 kbp, demonstrates a difficulty in determining which contigs *actually belong* to a haplotype in the sample. Later, I will show that the method presented in my thesis is capable of achieving superior results on this same benchmark, producing fewer, longer, gene-specific haplotypes, whose likelihoods correlate with their quality (Section 4.4).

Baaijens *et al.* stress the importance of not considering a prior reference, arguing that this approach will yield poorer recoveries. Yet **on the real HIV 5-strains dataset, the purely *de novo* formulation of** `SAVAGE` **that relies on an FM-index has worse performance than providing** `SAVAGE` **with an independent *de novo* assembly** of the sequenced reads. The latter constructs marginally fewer contigs, with marginally longer N50 and fewer mismatches, and is an order of magnitude faster. Additionally, their Discussion argues that an ad-hoc assembly is "often of worse quality than a well-curated reference sequence", yet their results on real data show that `SAVAGE` performs worse when the first step is conducted with a high-quality reference than a sample-specific ad-hoc sequence (or the FM-index). The work also describes support for paired-end reads, but the *de novo* method's use of an FM-index precludes the use of paired-end data, as the index structure destroys read pair information. The conclusions made in the manuscript do not appear to be supported by the application of `SAVAGE` to the only real data set.

The authors boldly state that "all existing assembly methods fail to address" the problem of distinguishing true mutations from sequence error, and/or addressing their reliance on some form of reference, and hail `SAVAGE` as "the first genuine de novo viral quasispecies assembly approach based on overlap graphs". Yet, their own work appears to perform better under consideration of a suitable reference, and its results on real data fail to provide insight on the quality of the hundreds of returned contigs in the context of how well they actually reconstruct the five original haplotypes.

2.7.4 PEHaplo (Preprint, 2018)

Earlier this year, Chen *et al.* posted a pre-print on their method **PEHaplo** [209]. Recognising the leverage that paired-ends and coverage (*e.g.* `EVORhA`; Section 2.7.2) afford the reconstruction of sequences, the authors propose a novel overlap graph that allows consideration of evidence observed on reads, their mates, and their coverage.

`PEHaplo` constructs a "paired-end read overlap graph" G_{PE}, where each vertex is a read fragment. G_{PE} maintains two distinct edge sets: E' where all pairs of mate reads r_{i1} and r_{i2} are joined by an edge, and E which connects a pair of reads r_i and r_j if a function $\delta(r_i, r_j)$ determines that the suffix of r_i overlaps the prefix of r_j over at least l positions. The edge is directed from r_i towards r_j and the length of the overlap is the edge weight. `PEHaplo` considers paths through E, directed by the edges in E'. That is, a path that begins at r_{i1} must also later contain r_{i2}.

The pipeline begins with a pre-processing step to drop low quality reads, or trim ambiguous base calls. Additionally, an alignment-based error correction tool is used to correct suspected read errors based on coverage information, and low abundance read data is discarded. Presumably these are all attempts to reduce the computational cost of constructing the overlap graph, rather than directly improving the haplotype recovery effort. Akin to `SAVAGE` (and originally, `HaploClique`), cliques in the graph are merged to form partial haplotypes, and the graph is further pruned to reduce complexity. Finally, haplotypes are formed by reducing the problem to finding paths through G_{PE} whose edges through E are best supported by the paired end evidence in E'. A depth first search (DFS) is used to find k maximum paths through the graph, similar to the approach of finding k disjoint paths with the polyploid-aware version of `HapCompass` (Section 2.6.1).

The work claims substantially improved results over `SAVAGE` on the *HIV* 5 strain mix: with significantly fewer, longer contigs and a lower mismatch rate. The work awaits peer-review.

2.7.5 Virus-VG (Preprint, 2018)

Baaijens *et al.* have recently followed up their work with **Virus-VG** [210]. Recognising that `SAVAGE` constructs many contigs that only represent partial haplotypes across the target genomes in the sample, `Virus-VG` considers the application of recent novel methodology for representing related genomes known as *variation graphs* (Section A.4.1) to join these contigs together. In a fashion similar to `EVORhA` (Section 2.7.2), `Virus-VG` considers frequency information across the contigs to guide the construction of maximal-length paths in the variation graph, to build full-length haplotypes. The results appear promising (and their Conclusions dismiss the abilities of another recently published viral quasispecies assembler [211]) but does not show application of the method to real data, and so fails to address questions raised by evaluation on real data in their prior work (Section 2.7.3).
The work awaits peer review.

2.7.6 Discussion

Given the small size of viral genomes, and their exceptionally high frequency of mutations, VQSR approaches can use more exact assembly methods than their SIH contemporaries, such as the construction of overlap graphs from whole reads rather than resorting to assembling a de Bruijn graph of k-mers. However, although whole read overlap algorithms appear to prove tractable for viruses, such approaches are more prone to construct false haplotypes due to noise in the reads creating spurious overlaps. Indeed, the algorithms presented do not assume a fixed number of haplotypes, but struggle to determine the true number of haplotypes, or provide a method for choosing between them.

Perhaps most interesting to discover throughout the survey of VQSR, is the lack of communication between the two communities working on SIH and VQSR. For example, Zagordi *et al.* wrote one of the first probabilistic VQSR methods, but failed to cite the work of Li *et al.* (Section 2.5.1), nor their later work on the probabilistic recovery of a diploid (Section 2.5.3). All VQSR works could have benefited with a definition of the SNP matrix M to unite notation and simplify descriptions of the problem. Despite its prevalence in the SIH literature, the matrix did not appear in any surveyed VQSR literature. Conversely, the mixture model presented in `MixSIH` by Matsumoto and Kiryu makes no acknowledgement of Zagordi *et al.*'s prior work either (Section 2.7.1). Indeed, despite an early literature review, I did not stumble upon the VQSR community until I had published my own pre-print and an online comment directed me to its existence[86].

[86] https://twitter.com/bioinformer/status/760628340273344516

2.8 Reconstruction of strains from a metagenome

2.8.1 Introduction

Our review has taken us from the very first definition of haplotype recovery (Section 2.2), through the early heuristics and the rise of minimum error correction as the *de facto* formulation that drove the majority of early implementations to solve SIH (Section 2.3). We've seen the construction of the first diploid human genome (Section 2.3.2), and the shift towards probabilistic methodologies as a more flexible and intuitive formulation to solve the problem of SIH (Section 2.5). We've explored early work towards polyploid approaches (Section 2.6), and the related but curiously not-overlapping problem of viral quasispecies recovery (Section 2.7).

Finally, we bridge the gap between the work surveyed so far and the matter at the heart of this thesis: metagenomes. I now turn our attention to work targeted specifically at shotgun metagenomic data. Like polyploids, the problem of recovering haplotypes from a metagenomic community has received limited attention so far, and current work remains in its infancy. As a greedy method that exploits overlaps to assemble sorted reads directly into haplotypes, I consider `Lens` (Section 2.6.3) a modern re-implementation of `FastHare` (Section 2.3.1): the first heuristic to address SIH in 2004. `Lens` ported the problem of SIH to the metagenomic research community, but was specifically designed for long-read datasets with low coverage. However, as mentioned in an earlier discussion (Section 2.5.9), current long-read technologies still suffer from high error rates and lower throughput than short-read alternatives [195, 196] that are still in favour when sampling from a metagenome [112]. Unfortunately for metagenomics, but not for my thesis, **no previously introduced work is suitable for application to deep shotgun metagenomic data, and as we will see, the current state-of-the-art is limited with regard to haplotype recovery**.

Researchers in the field have been quick to identify the inability of genomic assemblers to reconstruct strains from the metagenome [150, 152, 212, 213]. Indeed, the manuscript for `metaSPAdes`, one of the most versatile metagenomic assemblers (and arguably the *de facto* choice for metagenomic assembly), explicitly describes its focus "on reconstructing a consensus backbone of a strain mixture", rather than the strain-level variation itself [151]. With a lack of computationally tractable solutions from the emergent problem of SIH, the community at large fell back to simpler, generic methods.

Early work focussed its attention on taxonomic identification in metagenomic data, and formulated the problem as one of binning assembled reads or contigs by their tetranucleotide frequency [212, 213]. Noting in 2012 that there no efficient taxonomic profiling method existed for large metagenomic shotgun sequencing datasets, Segata *et al.* released (**MetaPhlAn**) [121], a tool capable of profiling millions of short-read sequences in minutes. `MetaPhlAn` maps reads with `BLASTn` to a reduced set of marker sequences that "unequivocally identify specific microbial clades at the species level". The tool determines species abundance by counting reads with sufficiently good hits to the marker database.

Notably, no preprocessing is required, as spurious reads should not have a hit to the conservative marker sequences. Segata *et al.* also introduced `PhyloPhlAn` [214], a database of universal markers for accurate reconstruction of phylogeny. Later, `Kraken` would become a popular alternative to marker-based tools for quick profiling, showing that it could assign more reads to a taxonomy than `MetaPhlAn` as it did not rely on taxon-specific marker genes, and it could do so an order of magnitude faster [120].

Many metagenomic binning techniques considered only nucleotide content, a methodology that remained largely unchallenged until 2014, when Alneberg *et al.* presented **CONCOCT** [215], a new approach to genome (species) binning that **considered both sequence composition, and coverage *across multiple samples of a microbiome***. The availability of multiple samples allows one to compare SNP frequencies between samples to identify and track the presence of strains. However, as I will come to describe, these approaches can only focus on single copy genes, as multiple copies would otherwise distort the frequencies.

These tools arose as a means to profile the constituents of a microbiome, but not to inform construction of their genomes or strains. But the novel consideration of sample coverage paved the way for alternative problem formulations that allowed the community to move towards initial work on strain-aware identification.

2.8.2 ConStrains (2015)

In 2015, Luo *et al.* identified that current **state-of-the-art approaches were "still limited to the species level"**, and that gaining assembled strains from a genomic assembler is a "rare exception", rather than the rule. As a means to identify the 'conspecific' strains of a species of interest and address this need, Luo *et al.* published *Conspecific Strains* (**ConStrains**) [216]. Guided by a single reference species, `ConStrains` attempts to infer strain-level structures in the population by identifying patterns of SNPs in the sequenced reads against the reference.

Their workflow begins by profiling the species-level taxonomy across all samples with `MetaPhlAn` (see above) [121]. The user must then select any species of interest for further analysis with `ConStrains`. For each of the chosen species, the corresponding set of marker genes from `PhyloPhlAn` [214] are used as a species-specific custom database, and the sequenced reads are re-mapped to those genes with `bowtie2` [217]. `ConStrains` discards alignments with orphaned mates, indels, low quality, or less than 95% identity to the reference gene. Additionally, subregions of genes with low coverage, or outside 1.5× the interquartile range of coverage for the gene, are masked out and ignored by `ConStrains`. Any gene masked over 30% of its length is dropped from the analysis. Variants are called naively over unmasked positions, where any site that observes an alternative allele at least twice is a SNP. For each species, the SNPs are considered as one sequence, concatenated over all

the reference genes for that species, and is referred to as the `uniGcode`. The goal is to determine the `uniGcode` for each strain.

The core approach behind `ConStrain`'s strain-level analysis is its *SNP flow* algorithm. For each species-sample pair, all the accepted SNPs are clustered by the Euclidean distance between the frequencies observed between their different alleles. This step groups SNPs together by the similarity of their frequency vectors. Clusters with fewer than 5% of the number of SNPs called for a species, or fewer than 10 SNPs total, are discarded. A directed graph links together the clusters, with weights defined by the observed frequencies. An exhaustive step considers all possible sets of paths through the clusters, to find the set of *SNP types* that minimises the difference (error) between the actual observed frequencies, and the frequencies that one would expect to see given a set of proposed paths.

The SNP-types across all input samples are then assembled by their sequence identity into a tree, and pruned exhaustively, with each cutoff representing a possible model for the candidate strains. `ConStrains` aims to minimise error between the true observed frequencies, and now those expected from the proposed strains in each pruned tree. To determine the strains that can explain the observed data, in both sequence and abundance, `ConStrains` employs Metropolis-Hastings MCMC[87] to efficiently sample the space of possible strain compositions and estimate per-sample strain abundance for each tree model. Finally, the sample-size corrected Akaike information criterion (AICc) selects the optimal proposal from the previous step.

The manuscript presents novel biological insight on previously published data, showing that it is **possible to identify strains and their abundance within a microbial community**, where work was previously limited to the species level. However, the approach relies on reference genes being available in `PhyloPhlAn`, and one is limited to reconstructing a `uniGcode` that consists of only unmasked regions of those genes, that feature sufficient, uniform coverage. Indeed, the word "haplotype" is avoided in the manuscript entirely. The authors identify that future work may wish to combine the approach with *de novo* assembly to overcome the need for any form of reference at all. This was later achieved by `DESMAN` (Section 2.8.4), whose authors were unable to compare their approach to `ConStrains` due to an issue with sufficient read coverage for the marker genes.

2.8.3 StrainPhlAn (2017)

Last year, Truong *et al.* (with Segata corresponding) highlight that we still lack the tools to perform "comprehensive strain cataloging" and that we are still not yet at the level where one can make inferences and comparisons between metagenomes in the same way that we can with isolate genomics. To move forward in this regard, Truong *et al.* introduce **StrainPhlAn** [218] as an extension of `MetaPhlAn`. `StrainPhlAn` continues the theme of 'fingerprinting' samples through the use of marker

[87] Gibbs sampling, which features in many of our surveyed approaches, is actually a special case of Metropolis-Hastings MCMC.

genes, essentially porting the underlying idea of ConStrains into MetaPhlAn itself, with the goal of inferring phylogenetic structure of strains across metagenomic samples.

StrainPhlAn works in a similar way to ConStrains (Section 2.8.2). Raw sequenced reads are mapped against the species-specific markers in the new version of the MetaPhlAn database [219]. Like ConStrains, the alignments are conservative, with many post-processing steps ensuring there is sufficient alignment coverage and quality. Strain profiling is only conducted on species that have at least 80% of their corresponding data markers spanned by valid alignments.

Unlike other fingerprinting approaches, for each marker-sample pair, a consensus sequence is generated by taking the majority rule of the read alignments. The resulting set of consensus sequences over all samples for a marker gene is then aligned with MUSCLE [220]. Each of the multiple sequence alignments are trimmed, processed to mask regions of low coverage or ambiguous nucleotides, and concatenated to generate a canonical alignment of all genes for each of the samples. Finally, the edited MUSCLE alignments are processed with RAXML [221] to build a maximum likelihood phylogenetic tree.

It's important to note in the context of this thesis that StrainPhlAn **only aims to reconstruct the single dominant strain for each sample**. It is this that permits its novelty: using the generated tree to enable elementary comparative metagenomics across samples of a microbiome, which was previously not possible with other marker approaches. Though, like ConStrains, StrainPhlAn's resolution is limited to considering entries in the MetaPhlAn database whose regions are sufficiently covered by the read data available. Again, the read alignment and multiple sequence alignments steps involve many filtering criteria, and limit the number of variant sites for which the strains are recovered. Once more, the word 'haplotype' does not appear in the manuscript: this work sets out to allow comparisons of the most abundant strain, and the abundance of the major strain for each species, in each sample, rather than sequence reconstruction.

2.8.4 DESMAN (2017)

More recently, Quince *et al.* stated that "to realise the potential of metagenomics fully" it is important to work towards developing methods capable of resolving species and strains from metagenomic data. Reference-based approaches such as MetaPhlAn and PanPhlAn become unsuitable when the environment in question has limited or no strain-level reference sequences available. The authors also argue that recent clustering approaches still fail to address the problem of disentangling the variation inside the bins, suggesting these "metagenome-assembled genomes" (**MAGs**) are still an aggregation of closely related strains, with resolution limited by the chosen assembler.

To overcome this, Quince *et al.* introduce *De novo Extraction of Strains from Metagenomes* (**DESMAN**) [141]: combining the core idea of ConStrains which uses the observed frequency of variants across samples, and applying it to binned assembled contigs (MAGs), rather than high-quality references that are simply not available in the metagenomic community.

The starting steps are familiar given the surveyed literature: *de novo* assembly and read alignment. For assembly, their tutorial[88] recommends `MEGAHIT` [222]. Sequence reads from all samples are 'co-assembled' together. `DESMAN` is sensitive to the quality of the assembly, and selection of appropriate assembly parameters are important to acquire more accurate contigs, as opposed to chimeras. The authors note that N50 is not necessarily useful for quality comparison between assemblies, as longer contigs could be chimeras that confound `DESMAN`. Reads are aligned back to the assembly with `bwa-mem` [223]. Given the assembly and resulting alignments, a matrix describing the average read coverage for each assembled contig, over each input sample is constructed.

The next step clusters assembled contigs together. Although `DESMAN` is compatible with any binning method, the manuscript recommends the use of an algorithm that can take both composition and coverage into account, such as `CONCOCT`[89]. Additionally, the sample coverage matrix is already formatted to leverage `CONCOCT`'s features. `DESMAN` relies on good binning to identify whether assembled contigs match some target species for further analysis. For more complex data, binning can involve multiple esoteric steps. After clustering, the bins are assigned to a taxon (it is unclear exactly how this is performed), and bins that do not match to the species of interest are discarded. A new sample coverage matrix is constructed to describe the average coverage and nucleotide frequencies across each of the contigs in the bin(s) of interest.

To identify strains from the bin, `DESMAN` targets genes that are expected to appear in every strain of interest in single copy, referred to as *Single Copy Core Species Genes* (**SCSGs**). One can identify SCSGs for strains of interest by downloading complete genomes for the strains in question and searching them against the NCBI COG database[90]. Without suitable reference sequences – arguably the most likely scenario when handling metagenomes for which there are few fully-sequenced isolates – `DESMAN` falls back to consider a collection of 36 single copy genes known to be conserved across all species that was introduced by their prior work [215][91].

SCSGs, or generic SCGs (hereafter 'genes') are found in the binned contigs with `RPS-BLAST`, and a new data structure that describes the sample coverage and nucleotide frequencies over the hit regions, is constructed. It is expected that genes should have the same coverage profile across all samples. If the coverage for a particular gene on a sample deviates from the median coverage of that gene over all samples by some threshold, it is considered an outlier. Genes with outliers across at least 20% of samples are discarded, and are not used by `DESMAN` for strain identification.

For detection of variants, `DESMAN` conducts a likelihood ratio test on each position of a gene, to determine the likelihood of whether the position has a single true base (with some level of error), or that the alternative hypothesis that two true bases are present, given the aggregate nucleotide frequencies across all samples. This is similar to the Bayesian model section method for determining

[88] https://github.com/chrisquince/DESMAN/blob/master/complete_example/README.md
[89] Which was published by several of the authors of the `DESMAN` manuscript
[90] The database for Clusters of Orthologous Groups (COGs) is an attempt at phylogenetic classification of proteins [224]
[91] Though Ciccarelli *et al.* had found their own set of "universal" COGs previously [225]

polymorphisms introduced by Kim *et al.* in 2007 (Section 2.5.3). A novel EM-like procedure generates the probabilities that a nucleotide was sequenced in error, which are stored in an error matrix for consideration by the two hypotheses (akin to the one described by Li *et al.*; Section 2.5.1). The predicted variants and error matrix are updated until convergence. The likelihood for the alternate hypothesis is parameterised and the user must select an appropriate threshold for the minimal frequency required to determine a consensus base. A final filter selects positions that are predicted with high confidence, to reduce the computational complexity of linking the variants together into haplotypes.

DESMAN assumes that the frequencies of selected variants in a sample must arise from some fixed number of real underlying haplotypes. The likelihood of observing a series of variants is modelled as a product of multinomials[92], whose detailed description I leave outside the scope of review. Due to the computational complexity required to explore the parameters space to optimise this likelihood, DESMAN employs a Gibbs sampler to generate samples and infer the model's parameters, in a fashion similar to Kim *et al.*'s work (Section 2.5.3), and ShoRAH (Section 2.7.1). Each iteration of the sampler draws and updates the conditional posterior for the haplotypes, sequence error and nucleotide frequencies in turn. In practice, some simplifications are made to encourage faster convergence.

As noted, the model assumes a fixed number of true haplotypes. Their proposed heuristic method executes DESMAN multiple times, aiming to recover a different number of haplotypes. As the *rate* of decrease in the posterior mean deviance[93] becomes small, DESMAN concludes the major haplotypes have been recovered. Haplotype accuracy is inferred from multiple runs of DESMAN with the chosen number of haplotypes. Multiple runs yielding the same haplotypes with high similarity imply confidence in the result.

The manuscript goes on to describe DESMAN's ability to resolve the assignments of the "accessory genome" for each of the strains. As the number of strain haplotypes and their abundances are converged upon by the Gibbs sampler, DESMAN attempts to infer which non-core genes appear with each strain. Effectively it is assumed that the coverage of each non-core gene can be modelled by a sum of the strain abundances for which it appears on. In the context of this thesis, I limit my discussion to strain haplotype recovery.

DESMAN is first evaluated on a mock dataset of 20 genomes: 5 *Escherichia coli* and 15 others, with the goal to recover the 5 *E. coli* strains. The synthetic reads are assembled with MEGAHIT, and the contigs are clustered into bins with CONCOCT. Quince *et al.* separately identify 982 SCSGs for *E. coli*, but can only map 372 of them to the contigs contained in their two *E. coli* bins. I will later show that my method which uses the assembly directly – without pre-processing such as binning – was capable of mapping and recovering haplotypes for 814 of Quince *et al.*'s SCSGs (Section 4.5),

[92] A multinomial distribution can be considered as an extension of a Binomial distribution where a trial may have k possible mutually exclusive, exhaustive options.

[93] The mean of all statistical deviances $-2log(\hat{\mathscr{L}})$ for all t runs of the sampler.

indicating a weakness in DESMAN's reliance on binning. Over the regions they identify as variants, DESMAN achieves good results. Curiously, the online documentation does not seem to indicate how to obtain the resulting strain DNA sequences, and verification is a multi-step procedure involving multiple scripts. Later, I will show that my method is capable of similar results, but over considerably more sites.

Additionally, DESMAN was evaluated with a complex synthetic community of 210 genomes. Modelling an environment where taxonomic classifications were impossible, and SCSG collections were unavailable, performance was mixed. The authors attribute this to "failures of the species binning and mapping algorithms rather than the haplotype inference per se", although their haplotype inference algorithm necessitates the pre-processing steps, including the recommended use of their own binning tool, so this comment is somewhat perplexing.

In comparison to state-of-the-art, Quince *et al.* were unable to run ConStrains on this mock community, which reported insufficient coverage, despite the median coverage across the samples being well above the $10\times$ required.

Overall, the methodology is complex. The documentation describes many, many steps, with multiple external dependencies, esoteric intermediate file formats and glue code. The workflow is a collection of scripts, rather than a polished pipeline. DESMAN relies heavily on other tools, but its accuracy is especially affected by the quality of the contig binning, which is clearly a lossy process. The method discards genes with many outliers, and only considers SNPs for which it has very high confidence – missing some strain variation. Additionally, the manuscript shows that DESMAN performs best in the presence of many samples, with error rates increasing beyond 15% toward 30+% as the number of available samples falls below 10.

For identification of strains from real metagenomes, one is limited to the resolution afforded by the 36 single-copy genes. For each species of interest, one must bin the assembled contigs and run DESMAN many times to find the correct combination of parameters, including a prediction for the number of target strains. However, DESMAN **reminds us of the power of considering frequency information over multiple samples, providing a method capable of determining gene counts for the accessory genome**. Quince *et al.* also highlighted the ongoing importance of working towards a solution that can identify haplotypes from complex microbial communities.

2.9 Conclusion

Clearly the problem of determining haplotypes is an important and rich area of research, and has received much focus since it was first introduced in 2001. However, as we have seen, **the majority of current approaches are simply not designed, nor are they appropriate for metagenomic applications** (Sections 2.3 - 2.7). For metagenomes, the problem of haplotyping is yet to be formally defined in the same way that single individual haplotyping was introduced as a problem by Lancia *et al.* back in 2001 (Section 2.2). In Section 3.1, I will propose my own formalisation of the problem of recovering haplotypes from microbial communities.

Neither the state-of-the-art for strain reconstruction (`DESMAN`; Section 2.8.4), or viral quasispecies recovery (`SAVAGE`; Section 2.7.3), nor specialist metagenomic assemblers (`metaSPAdes`; Section 2.8.1) are currently capable of reliably recovering enzyme haplotypes for arbitrary genes in a metagenome. Work in this area is generally limited to naive binning approaches, that rely on the use of one or more references, or subsets of known marker genes that are built from such references (Section 2.8). Each of the works presented for metagenomic data show it is possible to *identify* strains from microbial communities, **but it is important to note for the context of this thesis that identification is not the same as recovery**. That is, although Section 2.8.1 has shown that strain-level analysis is possible, with each work going beyond the species-level and presenting valid, interesting, novel biological insight on new and previously published data, these approaches rely on marker genes from the `MetaPhlAn` or NCBI COG databases. These markers are known to uniquely separate species and their strains, which are themselves built from known reference sequences. In the absence of references, `DESMAN` relies on 36 genes known to exist in single copy, that are capable of universally differentiating strains in a community. These tools are effectively identifying the *presence* of these markers and using them as 'barcodes' for abundance estimation and tracking between samples, but they cannot recover the actual haplotypes for arbitrary non-marker DNA sequences.

With the exception of `Lens`, which is an inefficient greedy approach that was not designed to scale to large sequencing projects (Section 2.6.3), current tooling does not attempt to recover haplotypes for arbitrary regions of a metagenome. Current haplotyping methodologies are limited to diploid species, clonal communities or samples with well-defined genomes. Generally, existing methods have one or more of the following limitations which make them unsuitable for metagenomic analysis:

- they assume that the solution is a pair of haplotypes from diploid parents, and discard/alter observations until a pair of haplotypes can be determined [144, 186]
- they discard SNP sites that feature three or more alleles as errors [186]
- they can generate a unrealistically large number of unordered potential haplotypes [199, 226, 206]
- they consider a fixed number of haplotypes which is not easy to determine [197]
- they are too computationally expensive for high-depth short read data sets [182]

- they require a good quality reference genome [227]
- they can only recover haplotypes for particular markers [216, 218, 141]
- they are no longer maintained/are specific to certain data/cannot be installed [228]

Research is gearing up to further explore the strains within the microbial communities in and around us [111], yet the focus stubbornly remains on the question of "Who?" is in a microbial community, with research vying to identify the names of the species and strains that contribute to the metagenome. **In my next chapter, I will propose a change in thinking, and argue in favour of uncovering the population-level variation within genes that perform a function of interest, which are shared across a community, in a method that is agnostic to taxonomy**.

No current work has considered the problem of recovering individual haplotypes for a gene of interest as a means to inspecting the variation of specific functions in the microbiome. Sections 3.2 and 3.3 will introduce **Hansel** and **Gretel**, my computational framework to recover haplotypes from the metagenome that overcomes the above issues and:

- does not require high quality reference sequences
- does not assume a fixed number of haplotypes, and needs no *a priori* knowledge of the number of haplotypes
- makes no assumptions about or attempts to correct or discard the distribution of alleles at any variant site
- does not need to distinguish between error and real variation
- uses all available evidence provided by the raw sequence reads
- does not require any user-defined parameters
- does not require pre-processing such as binning, or trimming or error correction of the reads
- can confidently rank its own haplotypes with likelihoods
- can be executed on an ordinary computer
- has actually been verified *in vitro* (Chapter 5)

Sections 4.2 - 4.5 present my evaluations, showing that it is possible to recover haplotypes from the metagenome, and also outperform both `SAVAGE` and `DESMAN` at their respective problems.

My next chapter will go on to formally define the **metahaplome**: the set of haplotypes representing the isoforms of a gene of interest that coexist across different species and strains in a metagenome. **It is key to understand that the problem I am proposing has a different goal to that of strain or viral-quasispecies reconstruction**. This thesis formulates a different, previously undefined research question, and aims to change the way that we consider and explore the variation within a microbiome.

Chapter 3

Theoretical Work

Section 3.1

The Metahaplome

3.1.1 Introduction

Genomic research has gone beyond the construction of a consensus sequence to represent a species in favour of understanding the true diversity that exists within populations [100]. High-throughput sequencing has allowed researchers to inspect the distribution of quasispecies in a viral outbreak [102] and estimate human haplotypes in major international initiatives such as the HapMap project [101].

Indeed, our own bodies are host to microbial communities responsible for a multitude of functions that impact our health and wellbeing, contributing to our metabolism, and defences. Perturbations in our microbiota can alter our development, or function, as well as influence our likelihood of contracting disease [5]. Our skin, gut and the environments that surround us, are theatres for a microbial war between organisms that are constantly innovating to adapt to their environment, and compete with an arsenal of enzymes to fulfill their niche [229].

Despite their impact on health, and their potential as a novel source of industrially relevant enzymes and antimicrobials, the individual organisms that inhabit these communities are not currently well understood. We are currently hindered by the resolution afforded by short-read sequencing, and long read technologies fail to provide the depth and accuracy necessary to adequately differentiate between real variation and error in a microbial community. As a result, characterisations are generally limited to taxonomic assessments by specifically sequencing the 16S rRNA-encoding gene and comparing the results to databases of known bacterial sequences, or clustering assembled reads.

However, it is population-level genetic variation that drives competitiveness and niche specialisation in microbial communities [105]. Novel combinations of variants that arise in individuals (haplotypes) are filtered by natural selection so that those that confer an advantage are retained [230]. Indeed, it has been shown that despite variation in community composition in extensive sampling of gut microbiota between individuals, function was remarkably conserved [145].

If we are to truly understand the ecological interactions in the microbiome, **we must look beyond taxonomy, and the pangenome, to analyse the variation exhibited not just between species, or even strains, but by the community as a whole**. I argue that with our current mindset – our obsession with recovering separate and distinct genomes of species and their strains – we are missing out on the true sequence variation in microbial communities: *the genes themselves*. After all, if function is so resilient, surely we want to see what variation exists throughout a community, rather than limit study to particular taxons that it is found in?

To better understand and exploit these communities, we should be looking to recover the unique "fingerprint" of genetic variants that encode the subtle differences in function for a gene of interest, across a community. If one could recover the individual variants – isoforms – of an enzyme producing gene, this would advance exploitation of industrially relevant compounds for the refinement of biofuels, production of plastics, bio-remediation and even the identification of new classes of antibiotics [32]. I term this collection of gene-level haplotypes, the "**metahaplome**". I propose, that for a particular gene, we can recover the metahaplome by collating all evidence provided by the sequenced reads to analyse their joint diversity, regardless of taxonomic assignment. **With this mindset, we could begin to catalogue the variation within a single gene, across every genome that features that gene in a microbiome, and capture the population level varation within a community.**

As I introduced in my literature review earlier, information on this level of diversity is currently lost through *de novo* analysis pipelines. Genomic assemblers aim to generate consensus sequences that discard the variation from individual members of the community (Section A.4). Even enhanced assemblers designed specifically for metagenomes do not address the problem of resolving the variation between strains or closely related species (Section 2.8). Analytical methods for metagenomes are currently limited to tools that naively cluster and bin reads or assembled contigs (Section 2.8.1), relying on the availability of sufficient reference databases, and losing vital evidence of variation to bins that are not selected for further inspection.

Until now, our insight has been limited to the study of 16S-rRNA genes, or specific markers (Sections 1.5, 2.9). However, in this Section, I will provide the first formalisation of the problem of recovering isoforms of genes of interest from a microbial community. Although we cannot rely on the assembly process to be correct, I will argue shortly in Section 3.1.4 that the assembly provides a means to collect together sequenced reads that are likely to be involved in encoding the same function. By reframing the problem of uncovering variation from a community as one of recovering genes and their variants, rather than reconstructing genomes, we can not only begin to answer the important biological question of how a particular isoform of a gene impacts a community and its environment, but we also reduce computational complexity, reduce the reliance on the reference and remove the need for naive pre-processing steps such as binning.

3.1.2 A Formal Definition

We formulate the concept of the metahaplome, and the problem of recovering haplotypes from a microbial community formally. First we must provide the following definitions:

- Ω
  An environment of microbial organisms.
- O
  A set[1] containing each full genomic sequence, of each individual organism in environment Ω. O is *the* metagenome of Ω: encompassing all possible genomes in the environment.
- $o[i:j]$
  A sub-sequence $i..j$ of some genome $o \in O$.
- Gene g
  A known DNA sequence g, potentially responsible for the production of a protein capable of performing a catalytic reaction of interest, such as the hydrolysis of cellulose
- $\Delta(s, g)$
  Any function Δ that can determine whether a DNA sequence s (such as a sufficiently sized sub-sequence $o[i:j]$) has sufficient sequence similarity to g, such as `BLAST`. Δ may return a boolean, or a real value (*e.g.* expect value) that can be coerced to a boolean via a user-selected threshold.
- $\Gamma_g = set(\wr o_k[i:j] \mid \Delta(o_k[i:j], g),\ k \in 1..|O|,\ i, j \in 1..|o_k|,\ i < j, j - i \approx |g| \int)$
  One may collect a bag of sub-sequences across the elements of O, determined to have sequence similarity to g by Δ. Γ_g represents the set of unique sequences contained in the bag.

We define Γ_g as the **metahaplome** for the gene g, in the metagenome O. Consider g encodes a protein that performs some biological function of interest, then each $\gamma \in \Gamma_g$ is an **enzyme haplotype** of g. That is, generally, γ is a DNA sequence that encodes an isoform of g. Ideally, we wish to recover the set of all such haplotypes: Γ_g. Of course, the metagenome O is unknown, and our insight to O is obtained through environmental sampling and DNA sequencing. We must recover haplotypes from the available evidence, and require the following additional definitions to define the recovery of enzyme haplotypes from a metagenome:

- $\sigma = \text{Sample}(\Omega)$
  - A sample taken from microbial environment Ω.
  - $\sigma \subset \Omega$

[1] Arguably one could also consider this definition as a bag, where we could have duplicates of genomes in the environment. However, the later set builder definitions are less cluttered by restricting this to a unique set.

- M
  - A set[2] containing the genomes $m \in M$, for each individual captured in the sample σ.
  - M represents the metagenome that was captured in the sample σ, and is our insight into O.
  - M is not necessarily or likely to be representative of the entire metagenome O.
  - $M \subset O$
- $R = \text{Seq}(\sigma)$
  - R is the set of **reads** obtained from the sequencing of isolated DNA from sample σ.
  - A read consists of a sequence of nucleotide bases $r_i[j] \in \{A,C,G,T,N\}, i \in 1..|R|, j \in 1..|r_i|$.
  - A read describes a fragment of some genome $m_k[u:v] \in M$, with some degree of error.
  - Due to sampling bias acquiring σ from Ω, R is unlikely to be representative of the true genetic diversity in O.
  - Additionally, due to sequencing and PCR biases and error, R is unlikely to provide uniform and non-zero coverage of the residues across all $m \in M$.
- $C = \text{Assemble}(R)$
  - Contig set C (**Assembly**) constructed *de novo* from the reads R by some `Assemble` operation.
  - $c_i[j] \in \{A,C,G,T,N\}$ for $i \in 1..|C|, j \in 1..|c_i|$.
  - `Assemble` attempts to reconstruct M from the reads R, but typically fails to distinguish between highly similar sequences that should create distinct $c \in C$.
  - C poses as a **pseudo-reference** for the metagenome M.
  - Alternatively, C could be a set of high-quality reference sequences, if available.
- $A = \text{Align}(R,C)$
  - **Alignment** A, generated by aligning read set R to contig set C with operation `Align`.
  - $A_{c_k[i:j]}$ is the set of read alignments in A that cover any position between i and j on $c_k \in C$.
- $S = \text{Call}(A_{c_k})$
  - The set of genomic positions on a contig $c_k \in C$ determined to be single nucleotide polymorphisms (SNPs) by the operation `Call`, given A_c: the alignments of R against c_k.
  - `Call` may simply consider each 'column' $A_{c_k}[i]$ for $i \in 1..|c_k|$ and determine position i as a variant if there is a disagreement on the nucleotide at that position across the aligned reads. `Call` may also be a more complex variant prediction algorithm.

Our goal is to determine the metahaplome Γ_g, for some gene of interest g. Although M will not be entirely representative of O, it is the only evidence of the sequence diversity available to us. Thus we adjust our goal to instead find the most likely elements of Γ_g, given the evidence that can be derived from M, via the alignments A and variant sites S.

[2] Again, we refer to this as a set, rather than a bag.

3.1.3 Methodology

To enable recovery of a metahaplome from a metagenomic sample we require:

- g, a known **target** DNA sequence of interest to a user
- $c_k[i:j]$, a **region** of contig c_k, identified as having sufficient similarity to g by $\Delta(c_k, g)$
- $A_{c_k[i:j]}$, the **alignments** of the set of reads R against the contig region $c_k[i:j]$
- $S_{c_k[i:j]}$, the genomic positions determined to be **variants** over the region $c_k[i:j]$

A metagenomic assembly (which we refer to as a 'pseudo-reference') can be generated by assembling sequenced reads, with an assembler such as `Velvet` [231]. Using Δ, one may locate a gene of interest g, on a contig $c \in C$ by similarity search or gene prediction. We refer to gene g as the *target*. We want to recover the most likely haplotypes of g that exist in the metahaplome Γ_g, using the observations of co-occurring variants observed across the reads that align to $c_k[i:j]$.

A subset of reads that align to the target region can be determined using a short read alignment tool such as `bowtie2` [217]. Reads that fall outside the region of interest (*i.e.* reads that do not cover any of the genomic positions on c_k that are associated to the target g) can be safely discarded: they do not provide relevant evidence for the recovery of haplotypes on the region of interest.

Variation at single nucleotide positions across reads along the target, can then be called with a SNP calling algorithm such as that provided by `samtools` [232] or `GATK` [233]. However, to avoid loss of information arising from the diploid bias of the majority of SNP callers [186] (Section 2.5.6), our methodology aggressively considers any heterogeneous site as a SNP (see Section 4.1.1).

The combination of aligned reads, and the locations of single nucleotide variation on those reads can be exploited to recover real haplotypes in the metagenome: the **metahaplome**.

3.1.4 Assembly and pseudo-references

As discussed in Section A.4, genomic assemblers are typically designed for the assembly of single genomes, and as such are optimised to remove low level variation, and aim to produce a single consensus sequence. Software specifically aimed at the assembly of reads from a community tend to use sequence abundance information (typically discarded as repetitive regions by conventional assemblers) to partition sequence graphs and correct poorly assembled contigs, in an attempt to overcome some of the limitations of single genome assembly [234].

However, metagenomic assemblers do not aim to solve the problem of recovering haplotypes [151]. Whilst other researchers have identified the problem that consensus assembly poses for the downstream analysis and are moving towards alternative reference structures, such as graphical models (Section A.4.1), there is still no method for the recovery of individual haplotypes for regions of a metagenome.

For use-cases such as viral quasispecies reconstruction (Section 2.7), high-quality reference sequences are usually available, side-stepping the need to perform *de novo* assembly. Although `Gretel` does not force one to use an assembly if a high-quality reference is available, in the absence of such reference sequences – which is typical for environmental samples – we propose that a *de novo* assembly of reads from a metagenome can act as a **pseudo-reference**. However, this pseudo-reference is merely a consensus of the available read information, and therefore tools are unable to place reads with similar sequence but different taxonomic origin on separate contiguous sequences (contigs) accurately, constructing chimeric sequences that are unlikely to exist in nature (recall Figure A.4). The assembled DNA sequence for an identified gene is unlikely to constitute a viable protein. Metagenomic assemblies are thus unable to represent the true diversity that exists between and within genomes in the environment from which they are sampled. Despite this, typical analyses construct a metagenomic assembly, discarding the only evidence of the haplotypes available – the reads themselves.

Although unhelpful for haplotype recovery directly, I argue that one can use a pseudo-reference as a taxonomic-agnostic base against which we can collate reads that potentially share similar function, by aligning reads back to the assembly. In this regard, conveniently, an assembly offers a proxy against which to filter large read sets. This alignment operation alone will still not recover any of the true haplotypes that exist in the original sample. Short reads will only provide fragments of evidence if they cannot span an entire gene-sized region of interest, and longer reads are prone to high error rates which currently precludes haplotyping. However, given a region of interest on the pseudo-reference (found by sequence similarity searches or gene prediction algorithms executed against the assembly), I will show that we can leverage the evidence provided by the reads that align back to the region to guide the recovery of haplotypes.

3.1.5 Summary

Having formally defined the problem of recovering haplotypes from the metahaplome, the following sections will go on to introduce `Hansel` and `Gretel`: my computational implementation to solve this proposed problem. `Hansel` is the data structure used to store which SNP alleles co-occur with each other on the same read, for all reads in the alignment $A_{c_k[i:j]}$. `Gretel` leverages this data structure for the probabilistic recovery of the most likely haplotypes in Γ_g.

Recovering the metahaplome is not an equivalent problem to that of strain deconvolution, or quasispecies recovery. To be clear, this is a different, biologically relevant and unproposed problem in computational biology. The metahaplome defines the set of all haplotypes that correspond to a region of interest in a metagenome, such as a gene encoding an enzyme. Recovery of the metahaplome permits exploration of the gene-level variation observed on individuals of a microbial community, in a way that is not currently possible with taxonomy-oriented analyses (metataxonomics) or merely searching sequences for orthologous genes. The metahaplome presents an opportunity for a finer scale analysis compared to that of finding orthologous genes: we are not just talking about a

collection of genes between species, but the individual haplotypes of a gene that exist between and *within* species of a community.

The effectiveness of the metahaplome as a conceptual framework is influenced by the depth and quality of the sequenced reads (R), the choice of (pseudo) reference (C), the method used to find targets of interest on the reference (δ) and the read alignment (A). R can be controlled somewhat through one's sequencing budget, and the choice to define the metahaplome on a region of interest such as a gene or operon, rather than an entire genome circumvents a need for the reference C to be of high-quality (allowing for use of an assembly instead). However, the method for selecting regions on which to perform haplotyping (*i.e.* δ), and the choice of read aligner and parameters to produce A are somewhat less well defined, and will need some thought from a user. Generally, we rely on an annotation tool such as `prokka` [235] to determine regions on C for haplotyping[3], and a read aligner such as `bowtie2` or `minimap2` to align the read set R against C, adjusting read parameters (*e.g.* Section 4.3.3) to maximise the proportion of aligned reads as necessary.

Although haplotyping of larger regions such as an operon, or even global haplotyping of entire chromosomes or genomes is actually possible with `Hansel` and `Gretel`, given sufficient coverage of the target region, I argue that the recovery of enzyme haplotypes is a biologically interesting problem in its own right. We know that the functions carried out by these communities are robust to changes in their taxonomic composition [145], and gene-level variation in individuals of a community is the driver for niche specialisation [105]. Now, with a framework capable of recovering the haplotypes from individuals of a population, we are granted a more granular insight to the genetic variation in the microbial communities found in our bodies and the world around us.

[3] *i.e.* C is annotated by `prokka` and the resulting GFF is considered to be populated with regions where δ is positive.

Section 3.2

Hansel
A data structure for the storage of sequence variance

Hansel is a probabilistically-weighted, graph-inspired, novel data structure. `Hansel` is designed to store the number of observed occurrences of a symbol α appearing at some position in space or time i, co-occurring with another symbol β at another position in space or time j, where α and β are symbols of some alphabet Σ. For our approach, we use `Hansel` to store the number of times a symbol α at the i'th SNP of some contig c, is observed to co-occur (appear on the same read) with a symbol β at the j'th SNP of the same contig. `Hansel` is a four dimensional array[4] whose individual elements $H[\alpha,\beta,i,j]$ record the number of observations of a co-occurring pair of symbols (α_i,β_j).

3.2.1 Different from the typical SNP matrix

Our representation differs from the typical SNP matrix model (Section 2.2, [142]) that forms the basis of many of the surveyed approaches. Rather than an $m \times n$ matrix where rows are sequenced read fragments, and columns are indexed SNP positions, we do not store single reads but instead aggregate the evidence seen across all reads by position against the reference or assembly.

At first this structure may appear limited, but the data in H can easily be exploited to build other structures. Consider $H[\alpha,\beta,1,2]$ for all symbol pairs (α,β). One may enumerate the available transitions from position 1 to position 2. Extending this to consider $H[\alpha,\beta,i,i+1]$ for all (α,β) over i, one can construct a simple graph G of weighted possible transitions between all symbols (see Section 3.2.5 for formalisation of G). In our setting, G could represent a graph of transitions observed between SNPs on a genomic sequence, across all reads. Figure 3.1 shows how the Hansel structure records information about SNP pairs, and shows a simple graph constructed from this information.

Intuitively, one may traverse a path through G by selecting edges with the highest weight in order to recover a series of symbols that represent an ordered sequence of SNPs that constitute a haplotype in the metahaplome. The weight of an edge between two nodes may be defined as the number of reads that provide direct evidence for that pair of SNP values occurring together.

[4]For the sake of brevity and to match my pre-print and documentation, I will hereafter refer to this four-dimensional structure as a matrix. Strictly speaking, to be mathematically correct, it should be referred to as a rank-4 tensor or 4D array.

3.2.2 Different from a graph

Although the analogy to a graph helps us to consider paths through the structure, the available data cannot be fully represented with a graph such as that seen in Figure 3.1 alone. A graph representation defines a constraint that only considers pairs of adjacent positions $(i, i+1)$ over i. Edges can only be drawn between adjacent SNPs and their weightings cannot consider the evidence available in H between non-adjacent SNP symbols. By considering only adjacent-SNPs, one can traverse G to create paths that do not exist in the observed data set, as shown by Figure 3.2 (and Section 3.2.5). To prevent construction of such invalid paths and recover genuine paths more accurately, one should consider evidence observed between non-adjacent SNPs when determining which edge to traverse next.

3.2.3 Using information from non-adjacent SNPs, and the path so far

The `Hansel` structure is designed to store pairwise co-occurrences of all SNPs (not just those that are adjacent), across all reads. We may take advantage of the additional information available in H and build upon the graph G. Incorporating evidence of non-adjacent SNPs in the formula for edge weights allows decisions during traversal to consider previous visited nodes, not just the current node i.

That is, given a node i, the decision to move to a symbol at $i+1$ can be informed not only by observations in the reads covering positions $(i,\ i+1)$, but also $(i-1,\ i+1)$, $(i-2,\ i+1)$, and so on. Such a scheme allows for the efficient storage of some of the most pertinent information from the reads, and allows edge weights to dynamically change in response to the path as it has been constructed thus far. Outward edges between $(i,\ i+1)$ that would lead to the construction of a path that does not exist in the data can now be influenced by observations in the reads beyond that of the current node and the next. Our method mitigates the risk of constructing paths which do not exist.

The consideration and storage of pairwise SNPs fits well with the Naive Bayes model employed to simplify the potentially expensive calculation of conditional probabilities (Section 3.2.7). Although we describe `Hansel` as "graph-inspired", allowing edge weights to depend on the current path through G itself leads to several differences between the `Hansel` structure and a weighted directed acyclic graph. Whilst these differences are not necessarily disadvantageous, they do change what we can infer about the structure.

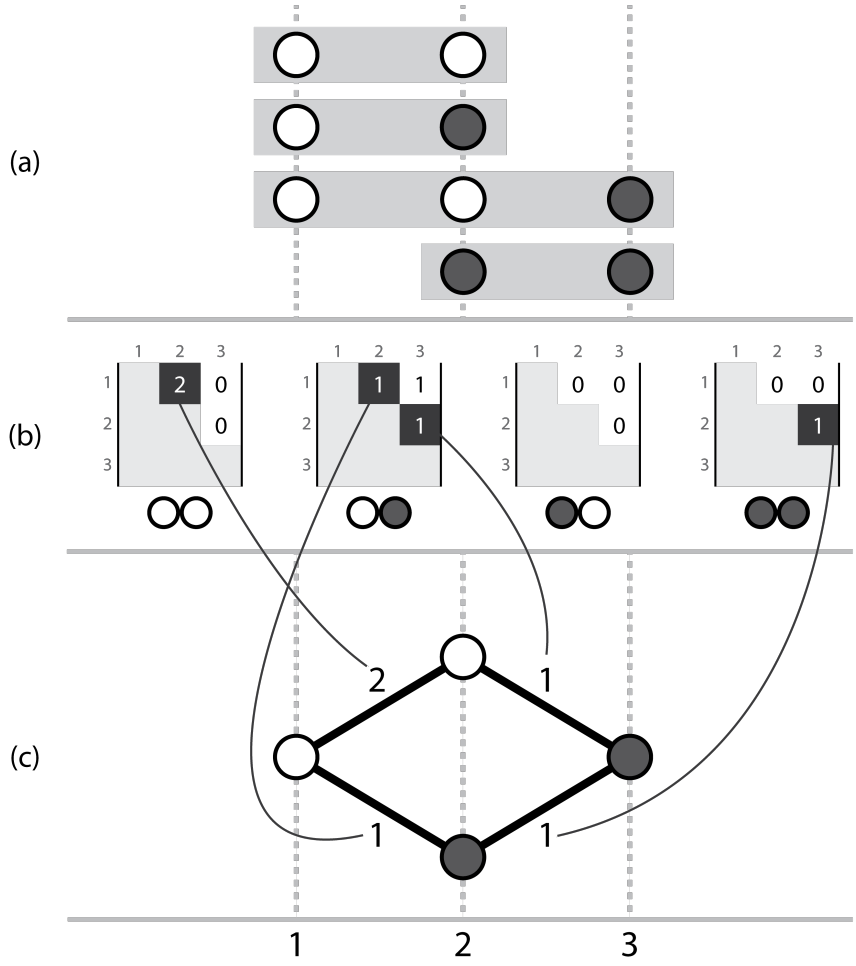


Figure 3.1 Three corresponding representations, (a) a set of short read sequences aligned to some (pseudo) reference, with called variants represented by white (0) and black (1) circles – the data structure discards non-SNP sites, (b) the 4D `Hansel` structure where each possible pair of symbols (00, 01, 10, 11) has a triangular matrix storing counts of occurrences of that ordered symbol pair between two genomic positions across all of the aligned reads, (c) a simple graph that can be constructed by considering the evidence provided by adjacent variants. Elements of the matrix used to determine navigational potential through the graph are shaded. Note this representation ignores evidence from non-adjacent pairs, which is overcome by the dynamic edge weighting (not shown) of the `Hansel` data structure's interface, described in Section 3.2.6.

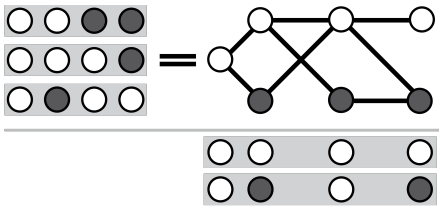


Figure 3.2 Considering only adjacent SNPs, one may create paths for which there was no actual observed evidence. Here, the reads {0011, 0001, 0100} do not support either of the results {0000, 0101}, but both are valid paths through a graph that draws edges between pairs of adjacent SNPs.

3.2.4 A dynamic structure

The structure of the graph is effectively unknown in advance. That is, not only are the weights of the edges not known ahead of traversal (as they depend on that traversal), but the entire layout of nodes and edges is also unknown until the graph is explored (although, arguably this would be true of very large simple graphs too). Indeed, this means it is also unknown whether or not the graph can even be successfully traversed.

Also of note is the fact that the graph is dynamically weighted. The current path represents a memory that affects the availability and weights of outgoing edges at each node. Edge weights are calculated probabilistically *during* traversal. They depend on the observation of SNP pairs between some number of the already selected nodes in the path, and any potential next node. Section 3.2.6 provides the equation and intuition for the probabilistic calculation of edge weights. Note that smoothing can also take into account 0 weighted edges between pairs of SNP symbols (α_i, β_j) that are not supported by the reads (Section 3.2.9).

We now have a data structure that permits graph-like traversal that is intrinsic to our problem definition, using informative pairwise SNP information collected from observations on raw metagenomic reads. `Hansel` fuses the advantages of a graph's simple representation (and its inherent traversability) with the ability to efficiently store pertinent information by considering only pairs of SNPs across all reads.

Although I describe `Hansel` as a dynamically weighted graph, one could argue that the paths we construct exist as a separate Markov chain with memory, which dictate how one can traverse a directed graph of adjacent SNPs (somewhat similar to the SNP graph of `HapCUT`; Section 2.3.5 and `HapCompass`; Section 2.4.2; but considering the actual symbols themselves). The graph acts as a 'lookup' of possible extensions to the Markov chain, and the underlying `Hansel` matrix is used to determine the most probable extension of the chain, given the path's memory. Both the path and the graph could be considered as separate but related components of `Hansel`. It is this encapsulation of both a Markov chain with memory and a directed graph that drives `Hansel`'s novelty.

3.2.5 Hansel as a graphical model

Consider an alphabet of symbols, Σ (*e.g.* $\{A, C, G, T, N, -\}$) and a list of n SNP positions $1..n$. Symbols $\varnothing_S$ and $\varnothing_E$ represent special sentinel positions at the start and end of the SNP positions (0 and $n+1$ respectively). The `Hansel` structure H can be considered as a graph $G = (V, E)$. Here, we define V, and E:

$$E = \bigcup_{i=1..n} \{(\alpha_i, \beta_{i+1}) \mid H[\alpha, \beta, i, i+1] > 0, \alpha \in (\Sigma \cup \varnothing_S), \beta \in (\Sigma \cup \varnothing_E)\} \tag{3.1}$$

$$V = \{v \mid (v, w) \in E\} \cup \{v \mid (w, v) \in E\} \tag{3.2}$$

E represents the set of edges, where an edge (α_i, β_{i+1}) is determined to exist in E if there exists at least one read whereby symbol α was observed at position i to co-occur with symbol β at SNP position $i+1$.

It should be noted, that although G can be constructed from H such that it is undirected and contains cycles, both properties lead to nonsensical haplotypes. Under such circumstances, `Gretel` could construct a path that visits multiple nodes that appear at the same i, or a trail that visits the same node multiple times. Such sequences would be meaningless in the context of haplotype construction, thus the interface to `Hansel` acts in such a way that G is a directed, acyclic graph.

We can define a haplotype as an alternating sequence of nodes ($v \in V$) and edges ($e \in E$). A path must always start and end at the special sentinel symbols $\varnothing_S$ and $\varnothing_E$, respectively.

$$\hat{h} = \varnothing_S, e_0, v_1, e_1, v_2, e_2, ..., v_{n-1}, e_n, v_n, e_{n+1}, \varnothing_E \tag{3.3}$$

Although, as only one directed edge between some v_i and v_{i+1} may exist, we can define $\hat{h}$ as a sequence of $v \in V$:

$$\hat{h} = \varnothing_S, v_1, v_2, ..., v_{n-1}, v_n, \varnothing_E \tag{3.4}$$

3.2.6 Probabilistic edge weights

If the construction of G only considers elements in $H[\alpha, \beta, i, j]$ where $abs(i-j) = 1$ (*i.e.* adjacent SNPs) it is likely one will recover haplotypes that do not actually exist.

Given the pairwise information available in H, for both adjacent, and non-adjacent SNPs, across all reads, edges in the graph G derived from H can be weighted probabilistically. We attempt to determine the next most likely symbol in a sequence, considering both the marginal distribution of symbols at the next position and the likelihood of those symbols appearing next, given an already observed partial sequence. That is, the next symbol v_{i+1} in a path depends not only on the current symbol (v_i) but some number of previous symbols ($v_{i-1}, v_{i-2} ... \varnothing_S$).

The outgoing edges from v_i are probabilistically weighted by exploiting the observations stored in the `Hansel` structure to create probabilities. These probabilities then determine the likelihood of moving from some v_i to each of the possible v_{i+1}.

We take a Bayesian approach to the problem of probabilistically weighting edges in `Hansel`'s graph representation. We define the probability of selecting v_{i+1}, conditioned on the path observed so far:

$$
\begin{aligned}
&\mathbb{P}(v_{i+1} \mid v_1, v_2, ..., v_{i-1}, v_i) \\
&\quad \propto \mathbb{P}(v_1, v_2, ..., v_i, v_{i+1}) \\
&\quad = \mathbb{P}(v_1 \mid v_2 ... v_{i+1}) \times \mathbb{P}(v_2, ... v_{i+1}) \\
&\quad = \mathbb{P}(v_1 \mid v_2 ... v_{i+1}) \times \mathbb{P}(v_2 \mid v_3 ... v_{i+1}) \times \mathbb{P}(v_3, ... v_{i+1}) \\
&\quad = \mathbb{P}(v_1 \mid v_2 ... v_{i+1}) \times \mathbb{P}(v_2 \mid v_3 ... v_{i+1}) \times ... \times \mathbb{P}(v_{i-1} \mid v_i, v_{i+1}) \\
&\qquad \times \mathbb{P}(v_i \mid v_{i+1}) \times \mathbb{P}(v_{i+1})
\end{aligned}
\tag{3.5}
$$

3.2.7 Simplification of conditional edge weights

Clearly, the number of factors in Equation 3.5 increases with i. For longer paths (more single nucleotide polymorphisms detected along the target region of interest), evaluating the equation becomes more computationally expensive, and risks potentially compounding estimation errors.

To construct a whole path p from $v_1 ... v_n$, the upper bound for the number of iterations will be $|\Sigma| \times n$ with calculations becoming increasingly complex as i increases.

To reduce complexity, we make an assumption of conditional independence between variants. Whilst this seems counter intuitive, the Naive Bayes model can deliver robust results despite its coarse assumption.

Thus we may simplify our previous equation and consider only the pairwise appearances of each v_i encountered thus far against v_{i+1}.

$$
\begin{aligned}
&\mathbb{P}(v_{i+1} \mid v_1, v_2, ..., v_{i-1}, v_i) \\
&\quad \approx \mathbb{P}(v_{i+1}) \times \mathbb{P}(v_1 \mid v_{i+1}) \times \mathbb{P}(v_2 \mid v_{i+1}) \times ... \\
&\quad = \mathbb{P}(v_{i+1}) \prod_{j=1}^{i} \mathbb{P}(v_j \mid v_{i+1})
\end{aligned}
\tag{3.6}
$$

However individual reads will not cover all SNP positions $1..n$ (if they did, we would not have to define the problem of haplotyping a metagenome by reconstructing sequences from reads). Thus, we need not consider all variants in the current path when evaluating edge weights. Instead, we could limit the number of variants to consider, from the current position in the path i, back some small and sensible number of steps L:

$$
\mathbb{P}(v_{i+1} \mid v_{i-L}, ..., v_{i-2}, v_{i-1}, v_i) = \mathbb{P}(v_{i+1}) \prod_{l=0}^{L-1} \mathbb{P}(v_{i-l} \mid v_{i+1}) \tag{3.7}
$$

Additionally, to overcome inaccuracies encountered through floating point error when performing mathematical operations on very small decimals, `Gretel` uses log probabilities instead. Via the log identity $log(ab) = log(a) + log(b)$ the product of the conditional probabilities becomes a sum of the log conditional probabilities:

$$log_{10}(\mathbb{P}(v_{i+1} \mid v_{i-L}, ..., v_{i-2}, v_{i-1}, v_i)) = log_{10}(\mathbb{P}(v_{i+1})) + \sum_{l=0}^{L-1} log_{10}(\mathbb{P}(v_{i-l} \mid v_{i+1})) \tag{3.8}$$

We define L as the the 'lookback' size, the number of variants of the current path to consider when selecting v_{i+1}. Conveniently, there is a reasonable intuition available for selecting a value for L: the mean number of SNP sites covered by the observed reads. Thus we avoid the scenario of introducing an algorithmically influential but difficult to optimize parameter, such as k-mer size for metagenomic assembly.

3.2.8 Estimation of probabilities

Equation 3.9 provides an estimate for the marginal distribution of a symbol β appearing at position j.

$$\begin{aligned} &\hat{\mathbb{P}}(v_j = \beta) \\ &\quad = \frac{\text{Number of reads with symbol } \beta \text{ at position } j}{\text{Number of reads spanning position } j} \\ &\quad = \frac{\sum\limits_{\gamma \in \Sigma} H[\beta,\ \gamma,\ j,\ j+1]}{\sum\limits_{\gamma \in \Sigma} \sum\limits_{\delta \in \Sigma} H[\delta,\ \gamma,\ j,\ j+1]} \end{aligned} \tag{3.9}$$

Equation 3.8 provides an estimate for the conditional distribution of symbol α appearing at position i given that β was observed at position j.

$$\begin{aligned} &\hat{\mathbb{P}}(v_i = \alpha \mid v_j = \beta) \\ &\quad = \frac{\text{Number of reads featuring } \alpha \text{ at } i \text{ and } \beta \text{ at } j}{\text{Number of reads spanning } i \text{ featuring symbol } \beta \text{ at } j} \\ &\quad = \frac{H[\alpha,\ \beta,\ i,\ j]}{\sum\limits_{\gamma \in \Sigma} H[\gamma,\ \beta,\ i,\ j]} \end{aligned} \tag{3.10}$$

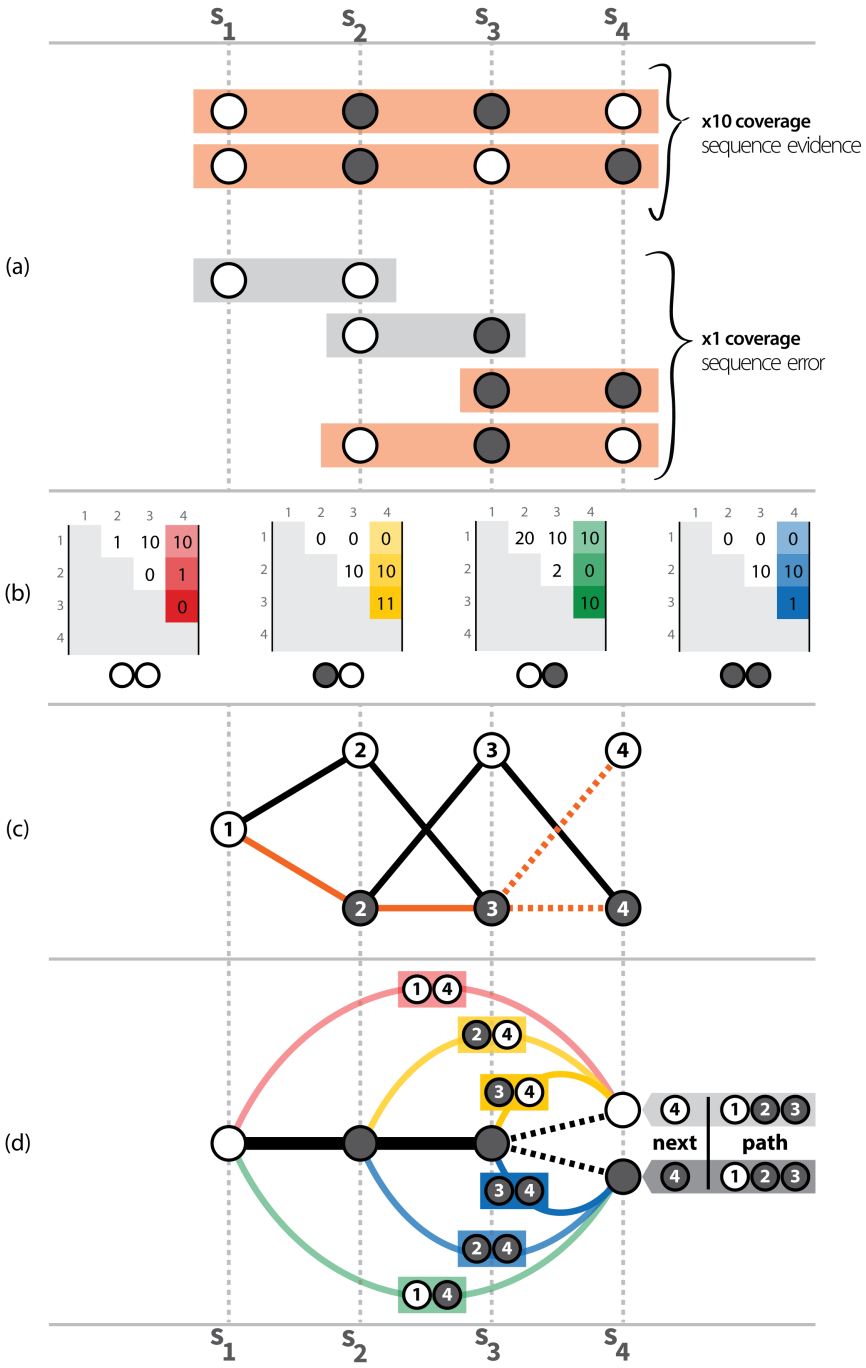


Figure 3.3 A detailed example demonstrating the relationship between sequenced reads and entries in the `Hansel`, and how those elements are used in the prediction of the next symbol in a path. (a) An example read set covering the first four SNPs (s_1 to s_4) of a hypothetical region of interest. For simplicity, the example uses binary elements (grey and white circles) rather than nucleotides. Two partial haplotypes are sequenced at $10\times$ depth, along with some errors at $1\times$ coverage, (b) The corresponding `Hansel` matrix for the example reads, (c) A graphical model of the `Hansel` matrix representing the navigable routes – but not necessary haplotypes (see Figure 3.2) – as defined by Section 3.2.5. A path is highlighted in orange, with a decision to be made regarding how to travel to s_4 (dotted orange lines), (d) Given the path so far $\{0, 1, 1\}$, the next element must be selected, using pairwise evidence stored in `Hansel`. Coloured lines match coloured elements in the `Hansel` structure in (b). Although not represented by the graph in (c), evidence of the highlighted pairs of co-occurring SNPs are used to determine the likelihood of moving to either of the two possible options.

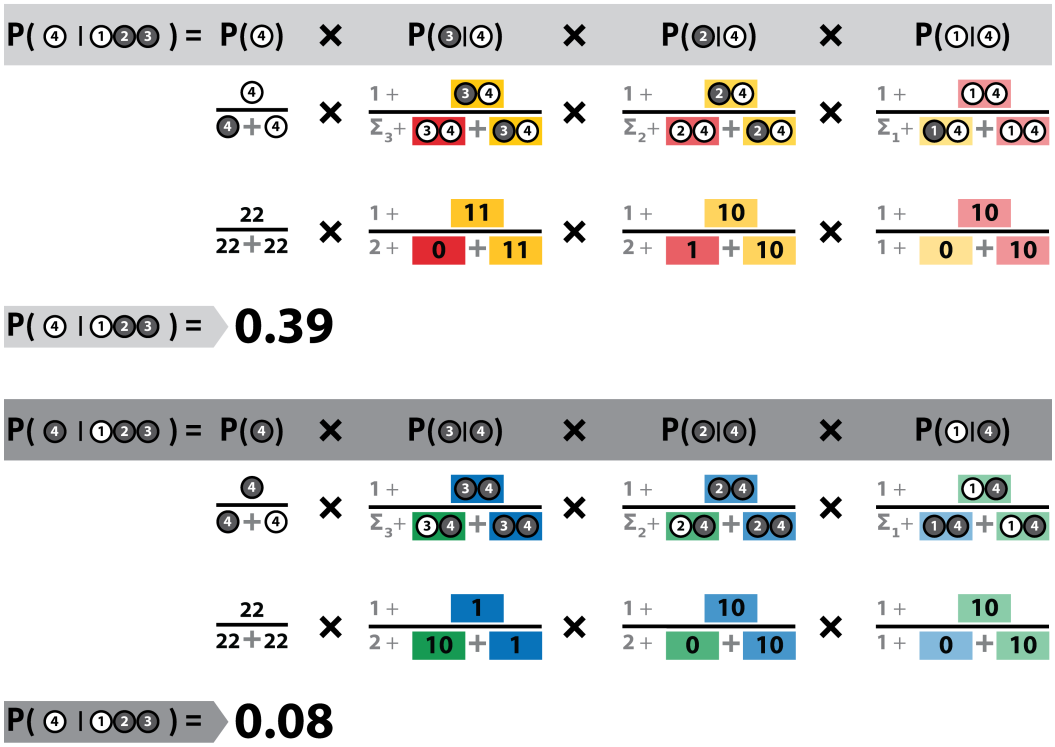


Figure 3.4 Calculation of the two edge weights corresponding to the example in Figure 3.3. The marginal distribution of $\mathbb{P}(s_4)$ is estimated via Equation 3.9, and the conditional distributions via Equation 3.10. Smoothing is applied as per Section 3.2.9: the numerator of each conditional estimated is incremented by 1, and the denominator by Σ_i – the number of variants at position i (*e.g.*, SNP s_1 features only white circles in Figure 3.3). Coloured blocks correspond to coloured elements in the `Hansel` structure. Given the path $\{0,1,1\}$, and the `Hansel` structure; we determine that the most likely path is $\{0,1,1,0\}$.

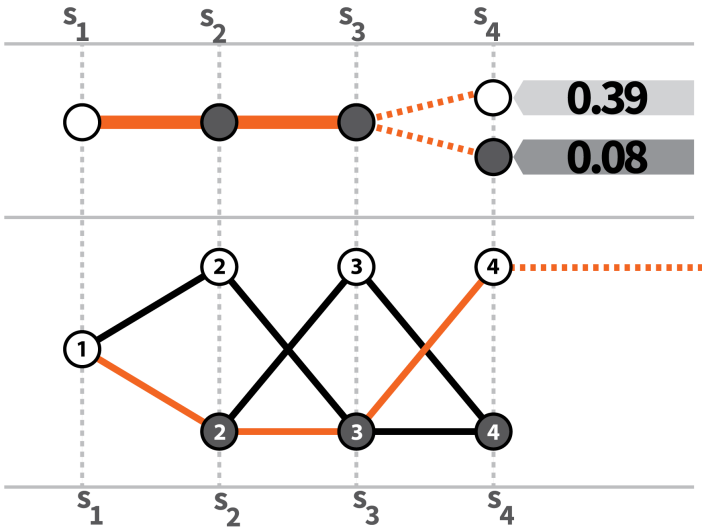


Figure 3.5 Selection of optimum next variant given edge weight calculations from Figure 3.4. (T) The path presented in Figure 3.3 has two possible options, with their likelihoods determined as shown in Figure 3.4, (B) The most likely next variant is chosen, and the path extended (highlighted in orange) to construct $\{0,1,1,0\}$.

3.2.9 Smoothing

To avoid the potential of dividing by 0 when using Equation 3.10 in cases where a suitable read spanning i featuring $v_j = \beta$ does not exist, we apply Laplace smoothing to effectively add a dummy support read. Future work will investigate alternative smoothing strategies.

$$\begin{aligned}
&\hat{\mathbb{P}}(v_i = \alpha \mid v_j = \beta) \\
&\quad = \frac{1 + \text{Number of reads featuring } \alpha \text{ at } i \text{ and } \beta \text{ at } j}{\text{Variants at } i \text{ + Number of reads spanning } i \text{ featuring symbol } \beta \text{ at } j} \\
&\quad = \frac{1 \; + \; H[\alpha,\; \beta,\; i,\; j]}{|\{\gamma_i \mid H[\gamma,\; \sigma, i, i+1] > 0, \gamma \in \Sigma, \sigma \in \Sigma\}| \; + \; \sum\limits_{\gamma \in \Sigma} H[\gamma, \beta, i, j]}
\end{aligned} \tag{3.11}$$

Note that this smoothing can only be applied if there is at least one read of support between position i and j for any pair of α and β. If no read covers both i and j, the graph will have no edges between those positions and it will not be possible to extend the Markov chain path to a SNP at j.

3.2.10 Summary

Here we introduced `Hansel`, a data structure for the storage of co-occurring SNP pairs that can support the construction of a graph that permits dynamic probabilistic exploration given the memory of any path taken through it. By not directly storing information for each of the reads and instead aggregating SNP pairs across all the reads, `Hansel` moves away from the traditional "SNP matrix" first introduced by Lancia *et al.* in 2001[5], and offers a new method for efficient handling and storage of variation observed across large read sets.

Figures 3.3 – 3.5 provide a detailed demonstration of how a set of reads can be represented by `Hansel`, and how the extracted pairwise evidence can be used to generate a graphical model that can be explored probabilistically for the reconstruction of sequences of symbols.

Although `Hansel` is presented here as a data structure specifically for storing pairwise SNP information, it is useful in its own right and can be used to hold other pairwise information such as basket items, or phrases; leading to uses outside bioinformatics in data analysis scenarios including affinity analysis [236] or natural language processing [237]. `Hansel` is provided independently at http://github.com/samstudio8/hansel or can be installed via `pip install hanselx`.

The following Section will introduce `Gretel`, an algorithm which will leverage `Hansel` for the recovery of haplotypes in a metahaplome.

[5]Which has been adopted by almost all haplotyping methods since (see Section 2.2)

Section 3.3

Gretel
An algorithm for the recovery of haplotypes from metagenomes

We introduce `Gretel`, an algorithm designed to interface with the `Hansel` data structure to recover the most likely haplotypes from a metahaplome. `Gretel` traverses the probabilistic graph structure provided by `Hansel`, selecting the most likely SNPs at each possible node (*i.e.* traversing edges with the greatest probability), given some subset of the most recently selected nodes in the path so far. At each node, an L'th order Markov chain model is employed to predict which of the possible variants for the next SNP is most likely, given the last L variants in the current path.

Execution of `Gretel` can be broken into the following steps:

1. Parse the read alignments and retain only the bases that cover SNP sites, discarding any conserved base positions as they provide no haplotype information.
2. Populate the `Hansel` structure with all pairwise observations from each of the reads.
3. Exploit the `Hansel` graph API to incrementally recover a path until a variant has been selected at each SNP position:
   - Query for the available transitions from the current position in the graph to the next SNP
   - Calculate probabilities of each potential next variant in the path given the last L variants
   - Append the most likely variant to the path and traverse the edge
4. Report this path as a haplotype and then remove the information for this path from `Hansel`'s data by reweighting the entries in the matrix ($H[\alpha, \beta, i, j]$) that contributed to this path. This will allow for new paths to be retrieved next.
5. Repeat (3-4) until the graph can no longer be traversed or an optional additional stopping criterion has been reached.

3.3.1 Greedy path construction

Haplotypes are reconstructed as a path through the `Hansel` structure, one SNP at a time, linearly, from the beginning of the sequence. At each SNP position, the `Hansel` structure is queried for the variants that were observed on the raw reads at the next position. `Hansel` also calculates the conditional probabilities of each of those variants appearing as the next SNP in the sequence, using a Markov chain of order L that makes its predictions given the current state of the observations in the `Hansel` matrix and the last L selected SNPs. `Gretel`'s approach is greedy: we only consider the probabilities of the next variant. Our razor is to assume that the best haplotypes are those that can be constructed by selecting the most likely edges at every opportunity, given the path so far.

3.3.2 Gretel's outputs

Finally, `Gretel` outputs recovered sequences as FASTA, requiring no special parsing of results to be able to conduct further analyses. Of course, with knowledge of the input haplotypes that we expect to recover, we are able to quantify our approach directly. For real metahaplomes, we assign each recovered haplotype a likelihood, to report our confidence in the returned haplotypes.

In addition to the sequences themselves, `Gretel` outputs a 'crumbs' file — a whimsical name for a simple, tab delimited format — that contains metadata for each of the recovered sequences. For each haplotype, the file defines the log probability of the initial `Hansel` matrix, given the haplotype's DNA sequence; how much of the evidence (*i.e.* raw observation counts) in the `Hansel` matrix support the haplotype, and the sum total of raw evidence that was removed from the `Hansel` matrix as a means to prevent the haplotype from being constructed again (detailed in the next section).

Currently, `Gretel` will continuously recover paths out of the remaining evidence until it encounters a node from which there is no evidence that can inform the next decision.

3.3.3 Reweighting to find multiple haplotypes

Whilst our framework is probabilistic, it is not stochastic. Given the same `Hansel` structure and operating parameters, `Gretel` will behave deterministically and return the same haplotype every time. Of course, we are interested in recovering the metahaplome: a set of real haplotypes, not just one. After a path has been constructed by `Gretel`, we manipulate elements of H to prevent repetitive generation of the same path and return the next most likely path on the next iteration instead. To do this, `Hansel` exposes a function in its interface for the reweighting of observations in the matrix.

Currently, `Gretel` reduces the weight of each pairwise observation that formed a component of a completed path - in an attempt to reduce evidence for that haplotype existing in the metahaplome at all, allowing evidence for other haplotypes to now direct the probabilistic search strategy. Given a path $\hat{h}$,

we inspect the marginal distribution of each element of the path in order to find the smallest marginal (λ). `Gretel` iterates over each element $\hat{h}[i]$ in the path, and uses the `Hansel` interface to reweight the element $H[\hat{h}[i], \hat{h}[i+1], i, i+1]$ by subtracting the result of multiplying the smallest marginal by the original value for that observation in H:

$$\lambda = \min(\{\mathbb{P}(\hat{h}[i]) \mid i = 1..n\}) \tag{3.12}$$

$$H[\hat{h}[i],\ \hat{h}[i+1],\ i,\ i+1] = H[\hat{h}[i],\ \hat{h}[i+1],\ i,\ i+1] - (\lambda \times H[\hat{h}[i],\ \hat{h}[i+1],\ i,\ i+1]) \tag{3.13}$$

The intuition is that the smallest marginal defines the least supported part of the haplotype, or the theoretical maximum proportion of evidence that all sites can agree as belonging to this specific haplotype. Each adjacent element of the matrix that contributed to the recovered haplotype has this maximum evidence proportion removed. Although, in practice the maximum value of λ is capped by `Gretel` in an attempt to stop aggressive reweighting that might otherwise prevent the recovery of closely related haplotypes.

3.3.4 Stopping criterion

After multiple iterations of path finding and subsequent reweighting, elements in H will begin to approach 0, causing edges in the graph to become unavailable for traversal. `Gretel` will immediately terminate upon encountering a node in the graph with no viable outgoing edges. That is, the selected symbol at the current i has no non-zero weighted edges to traverse between SNP positions (i, $i+1$) in the graph. Alternatively, if this criterion is not reached after 100 iterations (haplotypes), `Gretel` aborts.

3.3.5 Haplotype scoring

`Gretel` can score and rank the haplotypes it recovers. For a completed haplotype, $\hat{h}$, we compute its likelihood based upon the sum of the marginal log probabilities for each element of $\hat{h}$ given the current state of H.

$$\hat{h} = \varnothing_S, v_1, v_2, ..., v_{n-1}, v_n, \varnothing_E$$

$$
\begin{aligned}
L(\hat{h}) = \mathbb{P}(H \mid \hat{h}) &= \mathbb{P}(v_1 = \hat{h}[1], v_2 = \hat{h}[2], v_3 = \hat{h}[3], ...v_{n-1} = \hat{h}[n-1], v_n = \hat{h}[n]) \\
&= \prod_{i=1}^{n} \hat{\mathbb{P}}(v_i = \hat{h}[i]) \\
&= \prod_{i=1}^{n} \frac{\sum_{\gamma \in \Sigma} H[\hat{h}[i],\ \gamma,\ i,\ i+1]}{\sum_{\gamma \in \Sigma} \sum_{\delta \in \Sigma} H[\gamma,\ \delta,\ i,\ i+1]}
\end{aligned}
\tag{3.14}
$$

To overcome the potential for floating point arithmetic error (Equation 3.8), we calculate and report the log likelihood.

$$
log_{10}(L(\hat{h})) = \sum_{i=1}^{n} log_{10} \left(\frac{\sum_{\gamma \in \Sigma} H[\hat{h}[i],\ \gamma,\ i,\ i+1]}{\sum_{\gamma \in \Sigma} \sum_{\delta \in \Sigma} H[\gamma,\ \delta,\ i,\ i+1]} \right) \tag{3.15}
$$

3.3.6 Summary

We have introduced `Gretel`, an algorithm capable of using the `Hansel` data structure to store pairwise co-occurring SNP information for a set of aligned reads, and exploit this information to recover maximum likelihood paths through a dynamically weighted graph that represent haplotypes in a metahaplome.

`Gretel` is implemented in Python, and provides a command line tool to recover haplotypes from just a BAM and VCF. Haplotypes are returned in FASTA format (note that few other approaches actually do this) with a likelihood score that permits ranking and filtering.

`Gretel` is open-source and can be downloaded via https://github.com/samstudio8/gretel, or installed via `pip install gretel`, providing an easy-to-use command line tool, demonstrated in Listing 3.1.

```
gretel my_sorted.bam
       my_vcf.gz
       'test_hoot'
        -s 242 -e 512
        --master my_reference.fa
```

Listing 3.1 An example `gretel` command that attempts to recover haplotypes from the read evidence in `my_sorted.bam`, at the positions enumerated in `my_vcf.gz`, on the `test_hoot` contig, between positions 242 and 512. Homogeneous positions in the haplotypes (*i.e.* those not in the VCF) will be "filled in" by the master reference FASTA provided. Haplotypes are automatically written to `out.fasta`.

Chapter 4

In silico Evaluation

Section 4.1

In silico Testing Overview

This Section provides an overview of the methods to generate metahaplomes for both synthetic haplotypes, and haplotypes based on real genes; and the approach to evaluate the use of `Gretel`. Our test methodologies evaluate the performance of our framework against metahaplomes consisting of synthetic reads derived from both randomly generated haplotypes, and also haplotypes created from real gene sequences. Table 4.1 summarises each of the evaluation data sets.

Dataset Name	Length (bp)	Variation (SNPs/hb)	Read Sizes (bp)	Per-haplotype Depths	Replicates	Number of Read Sets
		0.001				180 × 5
		0.005				180 × 5
		0.01				180 × 5
`seq-gen`	3000	0.015	100, 150, 250	3, 5, 7, 10, 25, 50	10 × 5 trees	180 × 5
		0.02				180 × 5
		0.05				180 × 5
		0.1				180 × 5
DHFR	564	0.0696*	50, 150	3, 5, 7, 10, 25, 50	100	1,200
`DESMAN`	853.15†	0.12†	100	65.51†	1	814
HIV-gag	1500		219‡	6465.07‡		
HIV-pol	3009		217‡	9396.89‡		
HIV-nef	618	>0.2*	218‡	4211.87‡	1	1
HIV-vif	576		216‡	4822.18‡		
HIV-env	2568		214‡	4539.87‡		

Table 4.1 Properties of the *in silico* evaluation data sets. Per haplotype-base (hb) variation refers to the per-position mutation rate for a single haplotype in the metahaplome. Depth refers to the average number of reads that cover a single haplotype at each position. *Per haplotype-base variation rate for DHFR and HIV were estimated by dividing the ratio of SNPs over sequence length, by the number of known haplotypes. †Region length, variation rate and haplotype read depth for `DESMAN` data set are averaged across the 814 targeted regions. ‡Read size and depth for HIV-1 data sets were averaged over the region of the particular gene in the aligned BAM (see Table 4.5).

4.1.1 Read generation and variant calling

`shredder`: A simple read generator

For the `seq-gen` and *DHFR* data sets, reads are generated *in silico* with our Python based tool (`shredder`) which can be found as part of our evaluation repository via: https://github.com/SamStudio8/gretel-test. Our synthetic reads are designed to be simplistic; errorless and of uniform length and coverage. The synthetic read sets form a basis for testing the `Hansel` and `Gretel` packages during development, as well as providing a platform on which to investigate the influence that parameters such as read length, number of haplotypes, and mutation rate have on haplotype recovery.

For a given FASTA file, our tool generates reads of a uniform user-defined length and coverage, for each of the sequences in the file. The tool calculates the number of reads to generate to achieve the approximate coverage, given the length of the sequence, and the selected read length. A BED file can be used to mask particular areas of one or more of the input FASTA sequences. Uniform coverage is approximated by randomly generating the start positions of all of the reads across the input sequence (and also allowing for up to half of a read to extend beyond either end of the sequence).

As our tool is aware of the start position of every read that it generates, it is possible to also produce an alignment of those reads in SAM format. This allows us to align reads without introducing biases and assumptions from external tools.

```
python shredder.py --cover 20
                   --sam my_reads_align.sam
                   150 my_haps.fa > my_150bp_20x_reads.fq
```

Listing 4.1 An example `shredder` command that generates a set of 150 bp reads, at 20× coverage, from the sequences in the `my_haps` FASTA file, along with a corresponding SAM alignment.

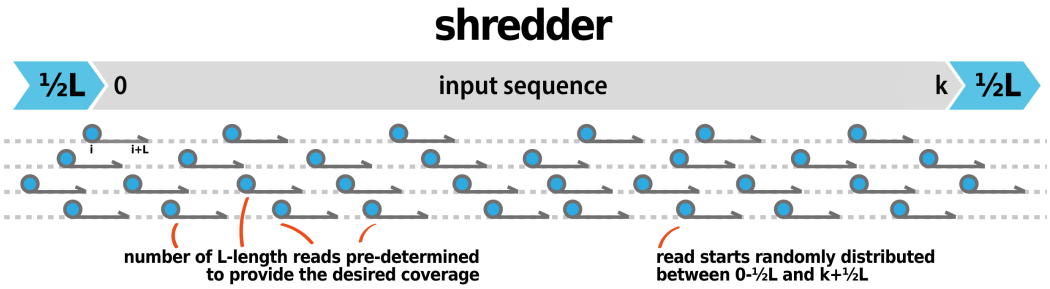


Figure 4.1 An illustration describing the intuition of `shredder`. Given an input sequence, desired read length (L) and coverage, `shredder` determines the number of reads (N) to meet the criteria. `shredder` generates reads by randomly selecting N start positions uniformly between the start and end of the input sequence, additionally allowing a truncated read to start up to a half-read before, or end a half-read after the input sequence. For each start position *i*, a read is created by copying L corresponding nucleotides from the input sequence.

`snpper`: An aggressive and naive variant caller

Pileups of our generated reads typically feature many tri- or tetra-allelic sites (especially as mutation rate increases), and we have seen that many variant calling approaches have a diploid bias, discarding such sites as sequencing error (Section 2.5.6). As our approach is robust to noise arising from sequencing error (*e.g.* performance on real HIV data: Section 4.4), we choose to aggressively call for variants from a set of aligned reads, by assuming any heterogeneous site is a SNP. Although naive, this ensures real variation is not unnecessarily discarded as error by external SNP calling tooling. Note that errors and misaligned reads have little co-occurring evidence in `Hansel` and so yield poor probabilities, meaning that `Gretel` is unlikely to incorporate these choices in recovered haplotypes.

Our evaluation repository contains the simple `snpper` tool that generates a VCF for a given BAM. `snpper` outputs a VCF record for any heterogeneous site. The code, documentation, and data for evaluation are open source and freely available via our data and testing repository: https://github.com/samstudio8/gretel-test

```
python snpper.py test.bam 'my_hoot' 3000 > test.vcf
```

Listing 4.2 An example `snpper` command that generates a VCF for the first 3000 bp of the `my_hoot` contig on the `test.BAM` BAM file.

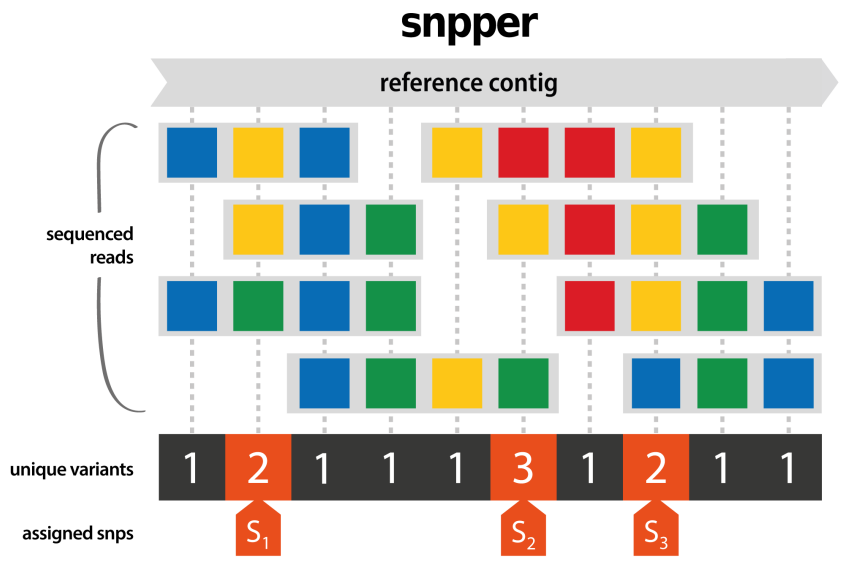


Figure 4.2 An illustration describing the intuition of `snpper`. Given a set of reads (coloured sequences inside grey boxes) aligned to some contig (from a reference or assembly), `snpper` checks the number of unique nucleotides at each genomic position. We assign any site without unanimous consensus as a SNP (orange).

4.1.2 Evaluating recovery accuracy

For each synthetic metahaplome, we perform a multiple sequence alignment with `MUSCLE` [220] to determine the definitive SNP positions. To evaluate the accuracy of a run of `Gretel`, each known input haplotype is compared pairwise to each of the recovered output haplotypes. Each input haplotype is matched to a corresponding "best" recovered haplotype. Best is defined as the output haplotype that yields the smallest Hamming distance from a given input haplotype (Figure 4.3). When calculating Hamming distance, we consider only the definitive `MUSCLE`-defined positions. That is, we exclude the comparison of homogeneous sites from the evaluation metric, to ensure we only consider our accuracy on positions that require recovery. For our results we report the proportion of total SNPs that were correctly recovered by `Gretel`, expressed as a percentage.

Comparing the definitive `MUSCLE`-defined sites, as opposed to the positions enumerated by `snpper` for each individual read set ensures `Gretel` is penalised when a SNP has not been called by `snpper`. Thus, `Gretel`'s performance can actually be better than the reported scores, but we aim to report the accuracy of the actual haplotype recovery effort.

Note that regardless of quality, all input haplotypes are assigned a best output haplotype. Additionally, an output haplotype may be the best haplotype for more than one input. For example, if only one haplotype is recovered by `Gretel`, it will be the best haplotype for all of the inputs, irrespective of the calculated Hamming distances. If more than one output haplotype has the same Hamming distance, the first that was found is chosen (this changes little in practice, but for some experiments we report the order in which best haplotypes are returned). If `Gretel` could not complete at least one haplotype (*i.e.* a pair of adjacent SNP positions were not covered by at least one read), all haplotypes are awarded a recovery of 0% – and are *not* excluded from the presented results.

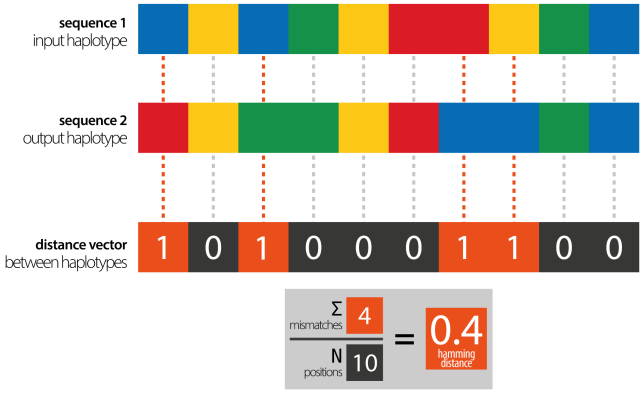


Figure 4.3 An illustration describing the concept of Hamming distance. A pair of DNA strings (coloured sequences) are compared. Each position is encoded in a binary string, where zero indicates a matching pair of symbols, and a one (orange) indicates a different symbol was observed. The distance is the sum of these values, divided by the number of positions, expressing the error proportion. This can also be expressed as "percentage accuracy", by subtracting the Hamming distance from 1 (worst possible score) and multiplying by 100.

Section 4.2

Experiment 1
Synthetic Sequences via Simulated Evolution

`Gretel` must first be tested on synthetic data to provide evidence of its capability. I chose to generate sets of haplotypes to serve as *synthetic metahaplomes*, and read sets with well-defined and controllable properties from which to recover the haplotypes. Later, Sections 4.4 and 5 demonstrate `Gretel` on real data from a HIV-1 sequencing experiment, and a real rumen microbiome.

This Section shows:

- The generation of synthetic metahaplomes with `seq-gen`
- The recovery of synthetic haplotypes with `Gretel` under varying conditions
- It is possible to recover haplotypes from microbial communities with short read sequencing
- A discussion of the successes and limitations on recovery from a community

4.2.1 Method

Initial experimentation with randomly generated haplotypes

For a robust evaluation of `Gretel`, I needed data sets where the properties of generated reads were known, well-defined and would not require external tooling for alignment and variant calling (such tools could confound initial evaluation). I first considered constructing sets of haplotypes as strings of equal length, with elements randomly selected from the alphabet `{A, C, G, T}`. Reads and a corresponding alignment could be generated from the haplotypes with `shredder` (Section 4.1.1). This was the methodology used to initially evaluate `Gretel` and is described as part of the first pre-print introducing this work [238].

Although this strategy afforded the ability to investigate the decisions that `Gretel` was making during recovery, ultimately the method was a poor proxy for the problem at hand. Simulations introduced in the original manuscript considered metahaplomes with hundreds of SNPs, and up to 25 randomly generated haplotypes with no intended phylogeny (*i.e.* little to no shared variation) — an unnecessarily difficult and highly unrealistic scenario. As the haplotypes were effectively strings of random SNPs, they exhibited low sequence identity between one another, which violates our definition of a metahaplome (Section 3.1).

It was necessary to construct a data set that permitted the constraining of properties such as read length, coverage and haplotype mutation rate, but with more realistic data, than random sequences. Still, as the majority of sites would be permitted to be tri- or even tetraalleleic, it was not suitable to use established variant callers, which typically introduce diploid bias [186], hence tools for handling metagenomic, viral or polyploid data typically employ their own variant calling (Sections 2.6 - 2.8). Section 4.1.1 describes my variant calling strategy, and its implementation: `snpper`.

I experimented with alterations to the random haplotype generation strategy, attempting to constrain sequences by some phylogeny, from an initial haplotype. However it was troublesome to produce sequences such that one could guarantee each had a fixed distance away from one another (a specific mutation rate) with a naive approach.

Simulating evolution

As an alternative strategy, `seq-gen` [239] is a tool designed to simulate the evolution of a nucleotide sequence along a given phylogeny. The phylogeny is provided via a `NEWICK` formatted **guide tree** (*e.g.* Listing 4.3), defining the course of evolution for the sequences to be generated, as well as the per-base **mutation rate** each of the sequences. In summary, the tool takes a DNA **start sequence**, and simulates evolution upon it, creating a new sequence for each branch of the tree and modifying the new sequences at the corresponding mutation rate.

For the purpose of testing my haplotype recovery approach, the goal was to use `seq-gen` to generate a synthetic metahaplome, *i.e.* a set of known haplotypes. Akin to the work already achieved with randomly generated haplotypes, sets of short reads would then be generated from the haplotypes and aligned back to the starting DNA sequence. Variants could then be called, and `Gretel` invoked to recover the known haplotypes, from the aligned reads.

```
((((A:0.001, B:0.001):0, C:0.001):0, D:0.001):0, E:0.001);
```

Listing 4.3 An example of the `NEWICK` formatted tree provided to `seq-gen` for the generation of synthetic metahaplomes. In this example, five "species" named `A` to `E` are specified with uniform branch lengths (*i.e.* the same base mutation rate, here: 0.001), to form a star shaped phylogeny.

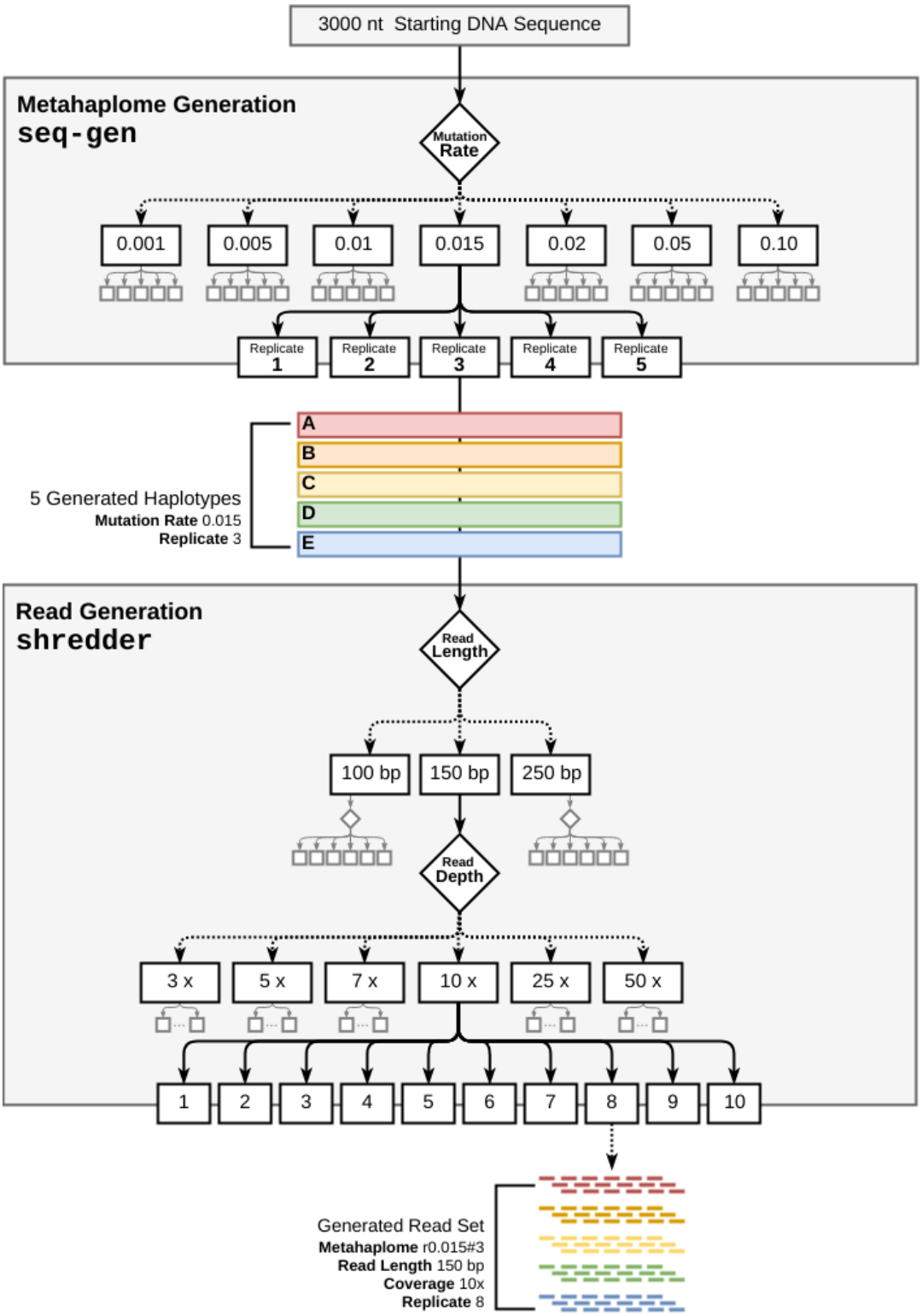


Figure 4.4 Hierarchy of read sets generated for the evaluation of `Gretel`. `seq-gen` was used to generate synthetic metahaplomes consisting of 5 haplotypes. The tool simulates evolution given a starting DNA sequence and mutation rate (*i.e.* number of nucleotide substitutions per site). 7 mutation rate levels, with 5 replicates yielded 35 metahaplomes. Metahaplomes were sampled for read generation at 3 lengths and 6 depths. Each length-depth pair had ten replicates, yielding 180 read sets per metahaplome and 6300 total.

For the creation of `Gretel`'s test data, the mutation rate of each of the haplotypes in a synthetic metahaplome were fixed to be equivariable (*i.e.* the branches of the guide tree were of uniform length), such that the haplotypes would be equally dissimilar to each other. This was both to ease the presentation of the analyses (as the mutation rate could be used as a factor to group results), but to additionally prevent a non-star phylogeny confounding the performance measurements of `Gretel`.

As part of this experiment, multiple metahaplomes were constructed in this way, but the number of haplotypes was always fixed at five. This was partly to reduce the computational resources required to generate, execute and analyse the data, but also to have a fixed variable against which other parameters could be allowed to change. It was important to inspect how other properties of the reads (*e.g.* read length, and coverage) would affect `Gretel`.

The same starting sequence was shared by all generated metahaplomes. A randomly generated sequence of 3000 nt with 50% GC content was used. The number of taxa in the trees was fixed at five, and the per-haplotype mutation rate was varied across seven levels. **35 different trees were generated** (7 mutation rates and 5 replicates) with `seq-gen`, each containing five sequences mutated at the same rate, from the original 3000 nt sequence. Each of the resulting 35 sets of five mutated DNA sequences represent a metahaplome from which the five haplotypes must be recovered by `Gretel`. Figure 4.4 describes the hierarchy of the generated metahaplomes and the derivative read sets.

As per the previously described read generation and variant calling protocols, synthetic reads were generated from the five sequences in a given metahaplome, varying both the read length and per-haplotype read depth (*i.e.* the average coverage of each haplotype). For each read length and depth parameter pair, ten read sets were generated, to amortise any effect on haplotype recovery introduced by the alignments of the reads themselves. For each of the 35 metahaplomes constructed with `seq-gen`, 180 read sets were generated (3 read sizes, 6 per-haplotype depth levels, 10 replicates), for **a total of 6300 read sets** (180 read sets, 7 mutation rates, 5 replicates). For the purpose of read alignment and variant calling, `shredder` could automatically generate an alignment against the 3000 nt starting sequence at the same time as the reads themselves (Section 4.1.1). Variants were called on the alignment with `snpper` (Section 4.1.1). Table 4.2 describes the average number of variants called across the 900 read sets generated for each mutation rate (180 read sets, 5 metahaplome replicates).

For each of the 6300 read sets, `Gretel` was executed to recover the five haplotypes from the generated reads. Figure 4.5 presents a flow chart outlining the steps to generate and analyse the synthetic metahaplomes. As described in Section 4.1.2, the method was evaluated by calculating the Hamming distance of all the recovered haplotypes, to the five known input haplotypes from the synthetic metahaplome as generated by `seq-gen`. We express the distance as a percentage, and report the average proportion of SNPs that were correctly recovered by `Gretel` across the five haplotypes.

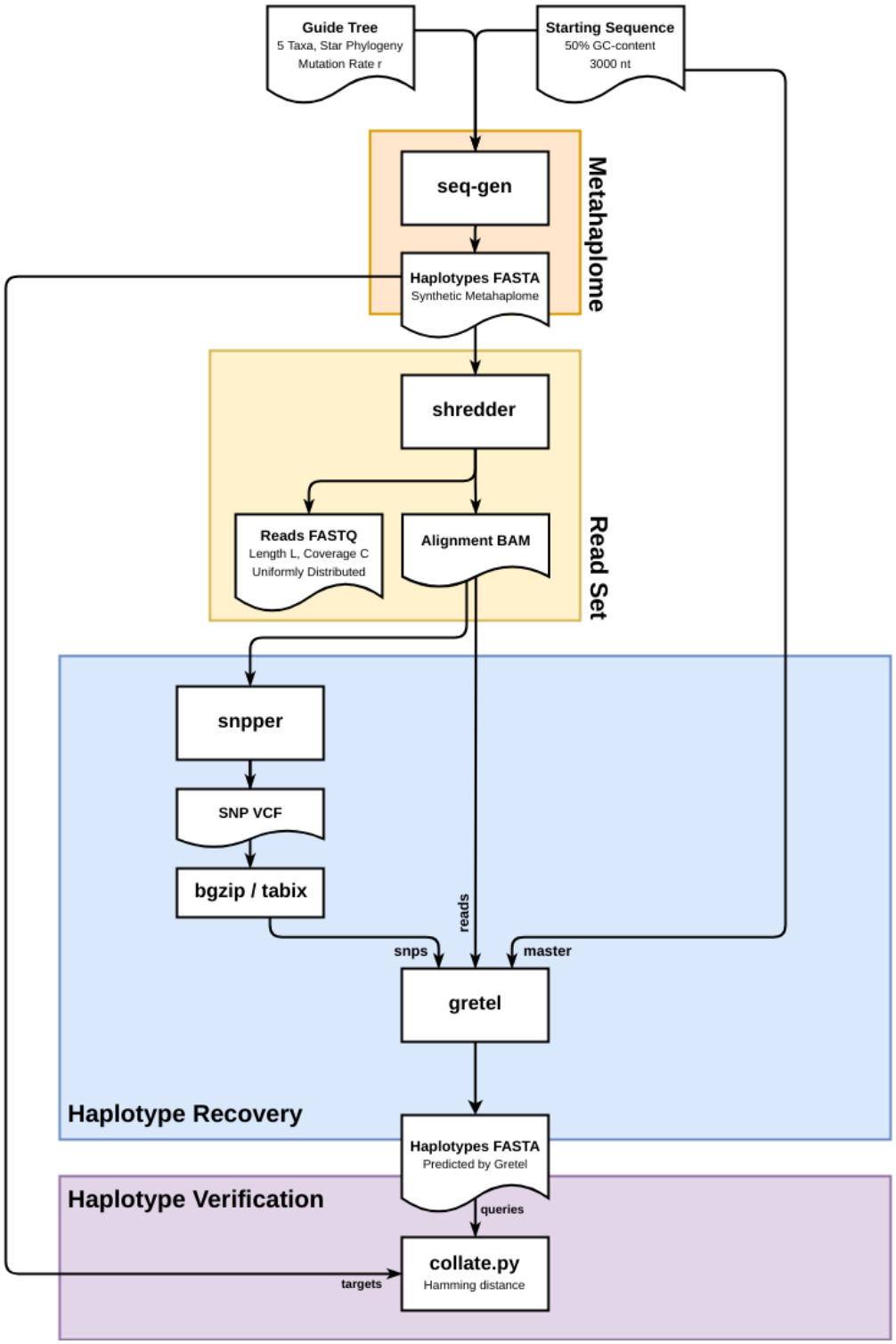


Figure 4.5 Flow diagram depicting the computational pipeline for the generation and analysis of a synthetic metahaplome. Note that `shredder` is capable of generating an alignment directly (Section 4.1.1), removing the need for an external alignment tool in the pipeline.

Mutation Rate SNPs/hb	Expected Mutation Rate SNPs/bp	Mean Called Variants SNPs	Effective Mutation Rate SNPs/bp (%)
0.001	0.005	17.25	0.0058 (0.6%)
0.005	0.025	71.20	0.0237 (2.4%)
0.010	0.050	141.54	0.0472 (4.7%)
0.015	0.075	209.25	0.0698 (7.0%)
0.020	0.100	277.28	0.0924 (9.2%)
0.050	0.250	640.63	0.2135 (21.4%)
0.100	0.500	1159.62	0.3865 (38.7%)

Table 4.2 Mean number of variants against the reference called by `snpper`, over the 900 synthetic read sets generated by `shredder`, for each per-haplotype (hb) mutation rate provided to `seq-gen`. Five replicate metahaplomes were constructed by `seq-gen` for each of the 7 levels of mutation rate. All metahaplomes were constructed with 5 haplotypes, allowing calculation of the expected mutation rate per bp of the metahaplome itself. As a single site can represent two distinctly evolved SNPs, the effective mutation rate is lower than the expected. The effective rate is also presented as a percentage of the 3000 bp sequences.

4.2.2 Results

Figure 4.6 presents the average percentage of variant positions correctly recovered by `Gretel`, across the 900 read sets (180 read sets across the 5 metahaplome replicates) for each read length-coverage parameter pair. Table 4.2 enumerates the mean number of variants called at each mutation rate, giving an approximation of the number of variants that must be recovered by `Gretel`.

We found that haplotype recovery improves with longer reads and greater coverage. We also observed potential lower bounds on our ability to recover haplotypes from a data set, as the facets with no successful recoveries show (0.005–0.05 SNPs/bp at 100 bp reads, 0.005 SNPs/bp at 150–250 bp reads). Unsuccessful recoveries are a result of at least one pair of adjacent variants failing to be covered by any read, which is a requirement imposed on `Gretel` for recovery. For shorter reads, low-level variation is more of a problem. 0.01 SNPs per haplotype base (hb) over 100 bp would yield just one SNP on average for each read - providing insufficient co-occurring evidence for `Gretel`.

Although one might expect high levels of variation to make the recovery of haplotypes more challenging, an abundance of variation actually provides more information for `Gretel`. We observe successful recoveries from data sets with high variation (0.1 SNPs/hb over five haplotypes of 3000 nt yields $\approx$ 1500 SNPs [Table 4.2]). With enough coverage ($\geq$ 7x per-haplotype depth), recoveries at a high level of variation are more accurate than those in data sets with fewer SNPs.

For realistic levels of variation (0.01–0.02 SNPs/hb) such as that observed in the human gut [135], a per-haplotype read depth of $\geq$ 7x permits the recovery of haplotypes with a median accuracy of 80%. At a per-haplotype depth of $\geq$ 25x, `Gretel` is capable of recovering haplotypes with 100% accuracy.

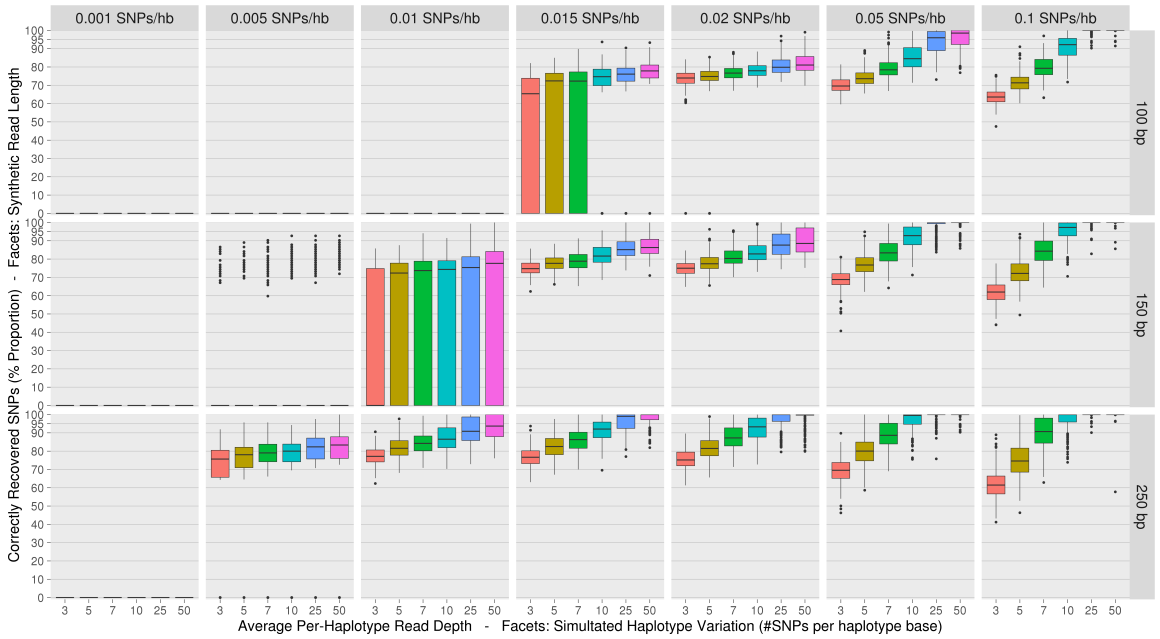


Figure 4.6 Boxplots summarising the proportion of variants on an input haplotype correctly recovered (y-axes) from groups of synthetic metahaplomes by `Gretel`. Single boxplots present recoveries from a set of five metahaplomes generated with some per-haplotype mutation rate (column facets), over 10 different synthetic read sets with varying read length (row facets) and per-haplotype read depth (colour fill). Each box-with-whiskers summarises the proportion of correctly recovered variants over the 250 best recovered haplotypes (yielded from 50 `Gretel` runs (5 metahaplome replicates $\times$ 10 read sets), each returning 5 best outputs). We demonstrate better haplotype recoveries can be achieved with longer reads and more dense coverage, as well as the limitations of recovery on data exhibiting fewer SNPs/hb. This figure may be used as a naive lookup table to assess potential recovery rates for one's own data by estimating the level of variation, with the average read length and per-haplotype depth.

4.2.3 Conclusions

Testing `Gretel` on synthetic data generated by `seq-gen` has allowed me to explore the boundaries for which the recovery of haplotypes is and is not possible. We see that generally, more coverage and perhaps counter-intuitively, more variation, improves our ability to accurately recover haplotypes from metagenomic data. Interestingly, when variation is observed at a rate greater than 0.015 SNPs per haplotype base, coverage appears to impact recovery accuracy much more than read length, this is probably because the length of a read now has enough SNPs to contribute pairs of SNPs to `Hansel`. I have identified that there are scenarios in which there is too little variation to provide evidence of co-occurring SNPs, preventing the recovery of haplotypes at all.

Of course, this experiment only considers haplotypes produced by computational evolution from a randomly generated DNA starting sequence. Although this was a superior and far more robust methodology than the generation of entirely random haplotype sequences – as random haplotypes have little to no shared variation which unnecessarily confounds recovery – I will now move on to test `Gretel` with sets of gene haplotypes that actually exist in nature.

Section 4.3

Experiment 2
Metahaplomes from real DHFR genes

To extend validation to data derived from real genes, and assess the robustness of the likelihood rankings of reconstructed haplotypes, I created a metahaplome consisting of five *dihydrofolate reductase* (*DHFR*) genes from multiple species and attempted to recover them from generated short reads.

This Section shows:

- `Gretel` is capable of working with sequences that exist in nature
- `Gretel` can work around indels and haplotypes that vary in length against the reference
- `Gretel` can recover sequences accurately, even when they diverge significantly from a reference
- Recovery rates are influenced by the availability of read evidence
- Likelihood scores provide a method to select the best haplotypes

4.3.1 Method

DHFR is an essential enzyme in nucleic and amino acid synthesis and is an important therapeutic target for infectious diseases such as malaria [240]. Identifying *DHFR* haplotypes can help researchers understand how sequence variation contributes to drug resistance.

An arbitrary *DHFR* gene was selected from GenBank (`EU145592.1`) to serve as the pseudo-reference against which to align reads for the purpose of variant calling. In a real metagenomic data set, one would use an assembly as a pseudo-reference to align sequenced reads against a genomic region of interest. It should be noted that the reference is not used by `Gretel` for the recovery of haplotypes, but is currently a necessary step to produce an alignment from which to call for SNPs.

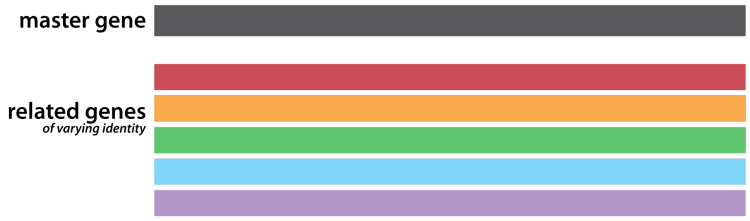


Figure 4.7 Five *dihydrofolate reductase* genes of varying identity (Table 4.3) were aligned against an arbitrary reference to create a synthetic metahaplome with five haplotypes to recover.

To find sequences to use as haplotypes, a discontiguous `megaBLAST` was conducted with the pseudo-reference. Five related but arbitrary genes of decreasing sequence identity (≈99.8%, 97.3%, 90.1%, 83.5%, 78.7%) were selected. Table 4.3 describes the selected genes. Although a set of increasingly divergent sequences from such different taxa may not appear to pose a "realistic" metahaplome, this experiment is designed specifically to observe the effect of distance from the pseudo-reference on `Gretel`'s recovery accuracy. As per our previously described read generation method, multiple sets of reads from the five input sequences were generated, with different levels of length and depth. Our data set is outlined in Table 4.1, and consists of 1,200 sets of reads (2 read sizes, 6 per-haplotype coverage levels, 100 replicates). `Gretel`'s goal was to recover the five input haplotypes, from the generated reads (Figure 4.8).

Unlike the methodology for the `seq-gen` simulations, the five input genes were not of the same length and could potentially contain insertions or deletions with respect to the chosen reference. The presence of such structural variation adds more difficulty and realism to the recovery problem in this experiment. As with a real data set, reads were therefore aligned back to the reference with a sequence aligner (`bowtie2`).

Variants were called on the alignment using the aforementioned `snpper` tool that determined all heterogeneous sites as variants. A multiple sequence alignment (`MUSCLE`) of the original five haplotypes showed the number of actual SNPs in the data set was 196. In general, `snpper` was able to call these 196 canonical sites as SNPs, with sufficient read evidence (Figure 4.9).

Accession	Organism	Length (bp)	Ref. Similarity (%)
EU145592.1	*Homo sapiens*	564	–
BC070280.1	*Homo sapiens*	564	99.823
XR_634888.1	*Papio anubis*	564	97.340
AK232978.1	*Sus scrofa*	564	90.071
M19237.1	*Saimiriine gammaherpesvirus 2*	552	83.514
XM_014960529.1	*Calidris pugnax*	551	78.662

Table 4.3 The chosen 'pseudo-reference' (`EU145592.1`) and the five genes that constitute the synthetic *DHFR* metahaplome. Genes with decreasing similarity to the reference were selected to pose a more challenging recovery problem. Figure 4.10 presents our recovery results for each gene.

Figure 4.8 Reads were generated from the five haplotypes with `shredder` and aligned against the reference with `bowtie2`. `Gretel` then attempted to recover the five original haplotypes from the aligned reads.

We evaluate `Gretel`'s performance by calculating the Hamming distance between the recovered haplotypes, and each of the five input genes. We express the Hamming distance as a percentage, *i.e.* we report the proportion of the 196 known SNPs from the multiple sequence alignment that were correctly recovered by `Gretel`. The evaluation method is described in more detail in Section 4.1.2.

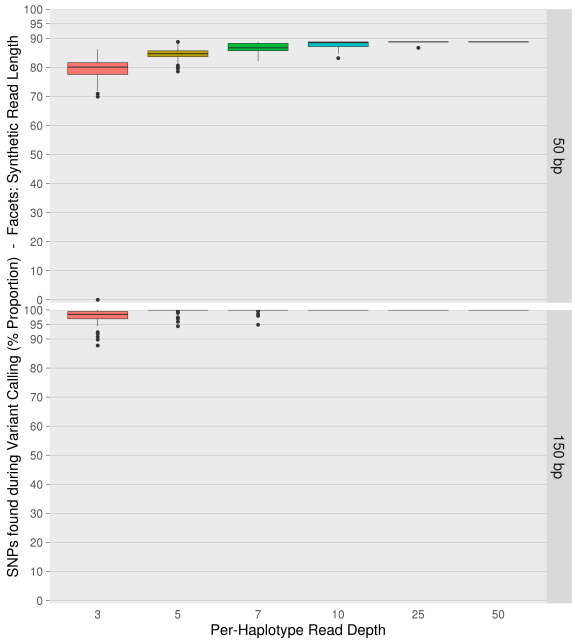


Figure 4.9 Proportion of the 196 canonical SNPs (called by `MUSCLE`) across the five *DHFR* haplotypes that were correctly discovered by our `snpper` tool (y-axes). Read sets are split by per-haplotype read depth (x-axis) and read length (row facets), each box-with-whiskers summarises 100 read sets. Uncalled SNPs prevent full recovery of one or more of the input haplotypes, due to insufficient evidence from 50 bp reads, or 3× coverage.

4.3.2 Results

Figure 4.10 presents the proportion of correctly recovered SNPs for each of the five input sequences listed in Table 4.3. `Gretel` achieves excellent recovery of *BC070280* and *XR_634888*, even with reads of just 50 bp. Recovery accuracy appears to be influenced by the similarity of the haplotype to the reference used. This is likely due to the availability of evidence as a result of read alignment, rather than a bias for the reference in `Gretel` (as previously described, the reference is not used during recovery by `Gretel`). Despite relaxed parameters yielding significantly improved alignment rates (Figure 4.12), there is still difficulty in recovering more divergent haplotypes, particularly for very short reads. Further analysis showed that this was due to insufficient read evidence being available for `snpper` to determine a site as heterozygous following read alignment (Figure 4.9), precluding the ability to attempt recovery at those sites with `Gretel`.

`Gretel` scores the haplotypes it recovers with a likelihood, representing the probability of the data observed in the `Hansel` matrix, given the recovered haplotype. To investigate the utility of the returned likelihoods, I calculated a "rank ratio" for each of the best haplotypes and checked its rank amongst the full, likelihood-sorted cohort of haplotypes returned in that run of `Gretel`. Rank ratio was calculated by sorting haplotypes by their likelihoods and assigning each a rank reflecting their position in the sorted set (starting with 0 for the best, 1 for the next, and so on). The rank indices were then divided by one less than the number of haplotypes returned, such that the best haplotype had a rank ratio of 0, and the worst a rank ratio of 1. The use of rank ratio here permitted comparisons across different runs of `Gretel`, regardless of the number of haplotypes returned, specifically for the case where our read sets with different parameters. However, using this mechanism causes loss of dimensionality by collapsing the distances between the returned likelihoods. I will later show (Figure 4.13) that considering scaled likelihoods is more appropriate in the general use-case.

Figure 4.11 plots the relationship between the accuracy of recovered haplotypes and this rank ratio. In general, the best reconstructed *DHFR* haplotypes can be ranked more highly than their peers. This property also improved as the similarity to the reference increased or when there was greater read length or depth. We show that our likelihood rankings are capable of differentiating best recoveries from closely related false positives.

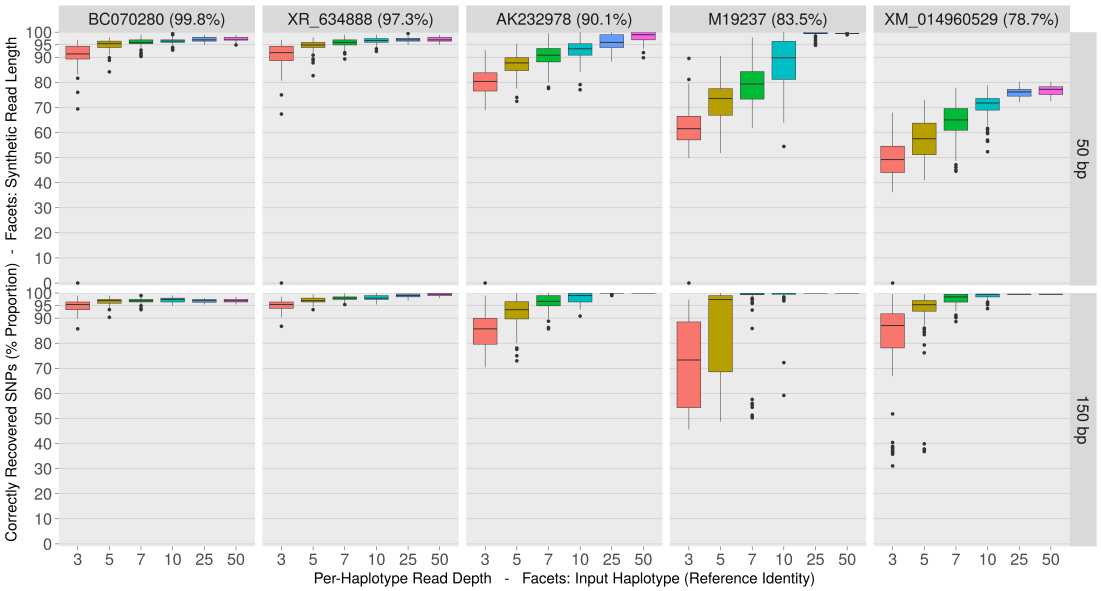


Figure 4.10 Boxplots summarising the proportion of variants correctly recovered (y-axes) for each input *DHFR* gene (column facets) by `Gretel`. We generated reads from the *DHFR* metahaplome at 6 different per-haplotype read depths (x-axes) between 3 and 50x, 2 read lengths (50 bp and 150 bp row facets) with 100 replicates. Individual box-with-whiskers summarise the recovery rate for a given gene, from reads of a per-haplotype depth and size, over the 100 replicate read sets. Input genes are sorted by decreasing identity to the pseudo-reference left to right. Note that there are a few cases where recovery is not possible at the lowest depth.

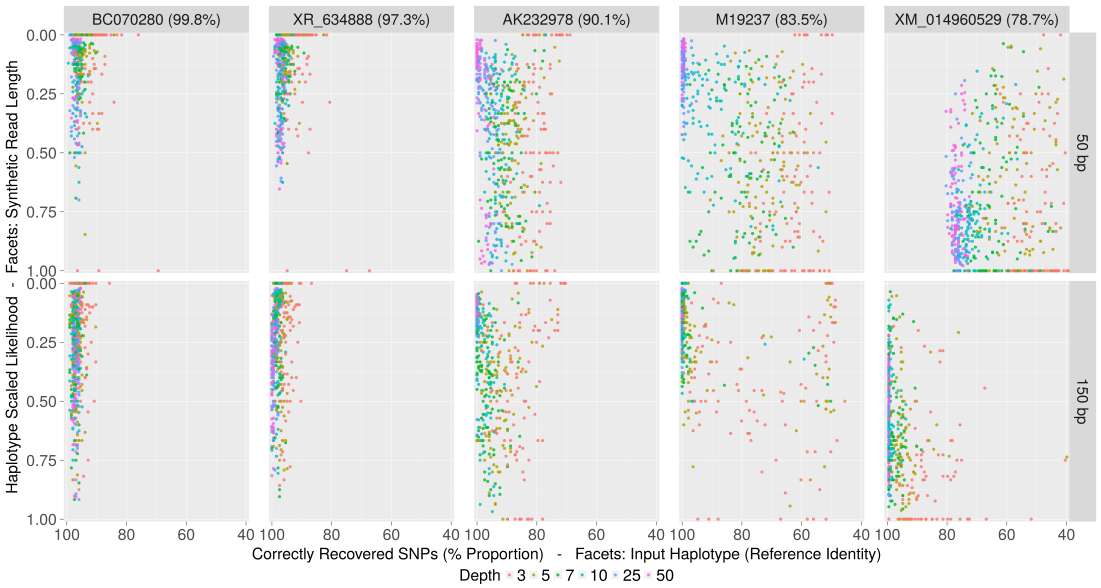


Figure 4.11 `Gretel` recovery rates (x-axes) against "rank ratio" (y-axes) for each gene of the *DHFR* data set (column facets) across the 1200 read sets. Haplotypes were ordered by their likelihoods (best to worst) and assigned a rank, beginning at 0. The ranks were divided by one less than the number of haplotypes returned for that run of `Gretel`. A rank ratio of 0 indicates the best (most likely) haplotype. We present recovery rates against rank ratio, separating the read sets by their read length (row facets) and coverage (colour fill and symbol). `Gretel` is capable of discerning accurate haplotypes and awarding better likelihoods to haplotypes with the most identity to real genes. Note the x-axes are truncated at 40%.

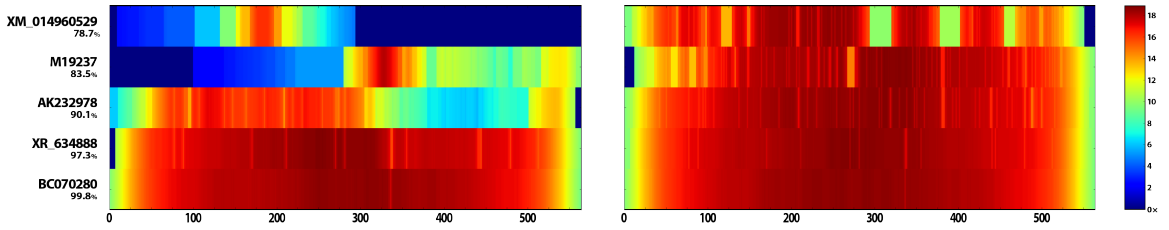
4.3.3 Initial alignments

As an aside, initial experimentation showed reads from the more dissimilar sequences were discarded by `bowtie2`. This is problematic, as it prevents relevant pairwise variant observations from being inserted into the `Hansel` data structure, yielding poor quality recoveries with `Gretel`. This was also true of other commonly used sequence aligners, including `bwa`. This was not unexpected, as such tools are not designed for use-cases where one wishes to permit such diverse sequences from widely different taxa to be co-aligned. However by significantly relaxing `bowtie2`'s parameters (Listing 4.4), I was able to improve overall alignment scores for the data set (Figure 4.12).

```
bowtie2 --local
        -D 20 -R 3 -L 3 -N 1 -p 8
        --gbar 1 --mp 3
        -x master.bti -U reads.fq --un unaligned.fq --no-unal -S out.sam
```

Listing 4.4 `bowtie2` parameters used to align generated reads to the pseudo-reference more permissively.

Note that I chose not to use these parameters in later experiments, such as the HIV-1 (Section 4.4) and *Escherichia coli* (Section 4.5) data sets, as the haplotypes were less diverse. The specific use-case here was to coerce the more dissimilar *DHFR* sequences (*i.e. XM_014960529* and *M19237*) to align to the pseudo-reference, where deletions were prohibiting straightforward alignment.



(a) `--very-sensitive-local` **(b)** `-D 20 -R 3 -L 3 -N 1 --gbar 1 --mp 3`

Figure 4.12 Heatmap of the average coverage for each of the five haplotypes across the alignment BAM files for all the generated read sets. The x-axis is the position along the pseudo-reference (1 - 564 bp) and the y-axes represent each input haplotype, ordered top-to-bottom by ascending sequence similarity. Tiles depict the average per-column aligned read count for a particular haplotype and genomic position. Low coverage in (a) highlights regions where `bowtie2` was unable to align the reads from more diverse haplotypes back to the reference under default parameters, causing data to be absent from the `Hansel` matrix, which was improved (b) by providing more specific parameters.

4.3.4 Conclusions

Here I used five hand-selected *DHFR* genes to observe whether `Gretel` was capable of discerning variation between closely related and divergent sequences mixed within a sample – somewhat emulating a metagenome. With sufficient evidence, `Gretel` recovers all five haplotypes well. However, I discovered that more divergent sequences are difficult to recover, as read evidence discarded by `bowtie2` caused some SNPs not to be called, leaving evidence missing from `Hansel`. Though it appears that this disadvantage can be overcome with longer reads or better coverage. Highly similar haplotypes (*e.g. BC070280* and *XR_634888*) can cause haplotype paths to converge, making it harder to disentangle real variation with 100% accuracy, warranting future work on `Gretel`'s reweighting approach (Section 7.3). Additionally, we observed that likelihoods are capable of choosing the best recoveries, though converting these to ranks loses information about their distribution. This experiment still used synthetic reads generated with an error-free naive approach. To address this, my next Section will apply `Gretel` to real sequencing data.

Section 4.4

Experiment 3
Recovering haplotypes from an HIV metahaplome

Following the successful application of `Hansel` and `Gretel` to synthetic data, I show in this Section that `Gretel` is capable of processing a **real** short read sequencing data set and recovering correct haplotypes. There are currently no metagenomic data sets which have both sequenced reads, and sets of known haplotypes. However, the related problem of viral quasispecies recovery (see Section 2.7) has a *de facto* benchmark data set consisting of five well studied HIV-1 strains mixed *in vitro* and sequenced on an Illumina MiSeq.

This Section demonstrates:

- `Gretel` is applicable to related haplotyping problems such as viral quasi-species recovery
- `Gretel` can scale to complex highly variable genes and thousands of reads
- `Gretel` is capable of recovering haplotypes from real short-reads with error and noise
- `Gretel` can recover haplotypes of highly variable viral genes with high accuracy

Strain	Length (bp)	*89.6*	*JRCSF*	*YU2*	*HXB2*	*NL43*
89.6	9669	100.00	–	–	–	–
JRCSF	9478	93.98	100.00	–	–	–
YU2	9659	94.05	95.10	100.00	–	–
HXB2	9719	94.43	95.18	95.71	100.00	–
NL43	9709	94.07	94.98	95.39	97.49	100.00

Table 4.4 Percentage identity matrix generated by `MUSCLE`, describing pairwise similarity between the five reference sequences corresponding to the HIV-1 strains contained in the *in vitro* benchmark mix.

4.4.1 Method

We apply our approach to a set of real reads consisting of five distinct HIV-1 strains mixed *in vitro* and sequenced on an Illumina MiSeq (Section A.2.2). The strains are presented in Table 4.4, along with their pairwise sequence identities as reported by `MUSCLE` [220]. Although the sequences of the five strains were known prior to mixing, the strains are likely to have mutated prior to the sequencing of the samples, introducing cryptic diversity [241]. Furthermore, the sequencing process can add additional noise through sequencing error. This provides a challenging data set for haplotype recovery. Sequence data can be accessed via ENA Run Accession `SRR961514`.

As per the previously established protocols for synthetic data, reads were aligned against a pseudo-reference. As high-standard reference sequences were actually available, one of the five strains was selected to serve as the pseudo-reference (Strain *89.6*). I chose to use one of the references as our pseudo-reference, as opposed to performing an assembly of the reads, as this experiment was designed to test `Gretel`'s haplotype recovery capabilities, where measures of accuracy could be confounded by the difficulties of *de novo* assembly. Reads were aligned against the reference with `bowtie2` (`--sensitive-local`). The overall alignment rate was 96.87%, yielding an alignment of 1,385,162 sequences.

Using the `snpper` tool, any heterozygous pileup in the resulting alignment was defined as a variant, resulting in a VCF containing 9,570 called variants. The SNPs are so numerous that they occur at 98.98% of all sites.

For the five longest genes on the HIV-1 genome (using *HXB2* gene co-ordinates [242]), I used `Gretel` to recover all haplotypes present, given the evidence observed across the aligned Illumina sequencing reads. The five genes and their associated read properties are presented in Table 4.5. Recovery with `Gretel` is straightforward; `Gretel` was executed once for each of the five genes, providing the same alignment BAM and VCF files, with `start` and `end` command line parameters to define the boundary of the particular region of interest. We evaluated our approach using the same framework as the synthetic metahaplomes in the previous Section. We report the sequence identity of all haplotypes recovered by `Gretel` against the five reference sequences, for each of the chosen genes.

Gene	Region (Size)	Average Coverage (×)	Called SNPs
gag	790—2289 (1,500 bp)	32,325.34	1,500
pol	2084—5093 (3,009 bp)	46,984.47	3,009
vif	5040—5616 (576 bp)	24,110.89	576
nef	8796—9414 (618 bp)	21,059.34	618
env	6224—8792 (2,568 bp)	22,699.35	2,568

Table 4.5 The five HIV-1 genes, HXB2 co-ordinates and properties of the aligned reads

4.4.2 Results

Table 4.6 shows the sequence similarity of the best matching recovered haplotype, for each gene and HIV-1 strain pair. In general, we recover the genes across the five different strains well. For each gene, at least one of the strains of that gene is recovered with 100% accuracy, with the exception of the protein envelope gene *env*. Though, `Gretel` is still able to recover one strain of the *env* gene with exactly one mismatch, with the DNA sequence of the recovered haplotype matching the reference at 2567 of its 2568 nucleotides, despite being the longest and most variable of the targeted genes. Bracketed values indicate the likelihood rank of the best haplotype amongst all returned haplotypes. With few exceptions many haplotypes feature in the top 10 likelihoods for each gene.

We hypothesise that the *env* gene had the most novel diversity, with the closest matching haplotypes to the original HIV strains only occurring in the top 25 most likely reconstructions. This is in contrast to *pol*, where the closest matching haplotypes to the original HIV strains were in the top 6 most likely reconstructions. This likely represents differing numbers of novel cryptic haplotypes of these genes and correlates well with their known mutation rates *in vivo* [243].

For all of the recovered HIV-1 haplotypes, Figure 4.13 plots the scaled `BLAST` bitscores against the `Gretel` likelihood score. Each recovered haplotype is matched to its closest strain. Our results show that ordering the recovered haplotypes by their likelihood scores can be used as a method to find the best candidates amongst the recovered sequences.

As reported in Table 4.6, haplotypes were discarded if they did not meet a conservative threshold of -1000 log_{10} likelihood. Manual inspection indicated that these discarded haplotypes showed a high number of deletions, an artefact arising from `Gretel` exhausting non-deleterious evidence in `Hansel` before terminating. Such haplotypes yield no significant `BLAST` hits against the NCBI `nr` database and our likelihoods provided a clear distinction between noise and useful recoveries.

4.4.3 Conclusions

This experiment applied `Gretel` to a more realistic data set, showing that the method is capable of handling data from a real sequencing experiment, with real noise and errors. My probabilistic approach is robust, as small numbers of erroneous observations have a low probability of being incorporated into the SNP chain.

`Gretel` performs remarkably well, recovering the correct identity of hundreds to thousands of SNPs (as `snpper` determined every position along each gene as heterogeneous) that compose the haplotypes. With the exception of the complex envelope gene, my method recovers at least one strain's exact sequence for each of the targeted genes; demonstrating its utility to the neighbouring viral quasispecies community (Section 2.7), who I would encourage to try out my method on their own data.

Figure 4.13 plots the scaled `BLAST` bitscore – used as a proxy here for haplotype quality – against the scaled likelihoods assigned by `Gretel`. We observe that better likelihoods correlate with better bitscores, indicating that we can reliably choose higher quality haplotypes for each of the five genes, by selecting those that have been assigned better likelihoods. Although I have chosen to use an arbitrary cut-off of -1000 to filter particularly poor haplotype candidates (featuring scattered deletions or stop codons), the likelihoods do appear to follow a sigmoid shaped curve, indicating a relationship which could better inform methods for haplotype selection. Future work for using likelihoods for haplotype selection and filtering is discussed later in Section 7.3. Although `Gretel` outputs many haplotypes, some of which have poor quality (low bitscores); the use of likelihoods overcomes this by providing a means to select promising candidates.

Beyond the alignment, `Gretel` does not require read processing, parameter bootstrapping or error correction. This Section has demonstrated that **it is possible to recover highly variable genes from short read sequencing of a real metahaplome, with 100% accuracy, and select the most likely haplotypes by ranking results by their assigned likelihoods**.

This experiment demonstrated `Gretel`'s capabilities on real sequencing reads, but still does not represent a real metagenome – a mix of species. My next Section will seek to recover haplotypes from a *de facto* benchmarking data set representing a mock microbiome.

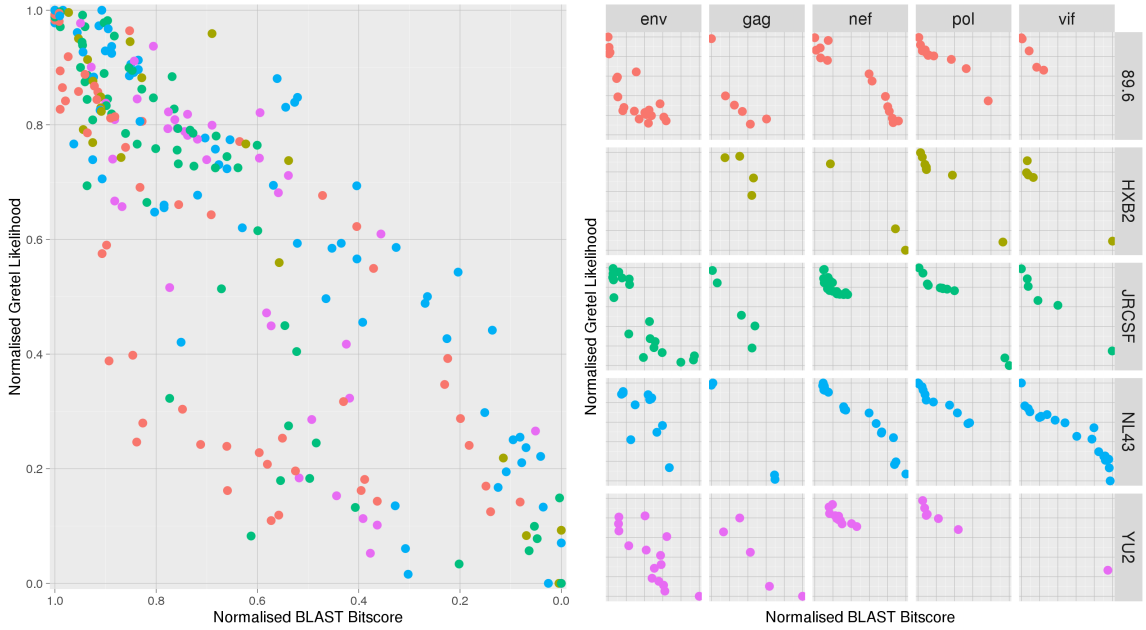


Figure 4.13 Scaled `BLAST` bitscores (x-axes) against scaled `Gretel` likelihoods (y-axes). (Left) Bitscores against likelihoods for all recovered HIV-1 gene haplotypes, coloured by strain, (Right) Plot facets show the bitscores against likelihood for all recovered haplotypes, separated by gene (column facets) and strain (row facets and colours). `Gretel` consistently awards higher likelihoods to recovered haplotypes that are better matches to a real haplotype.

			Strains				
Gene	**SNPs**	**Haplotypes†**	**89.6**	**HXB2**	**JRCSF**	**NL43**	**YU2**
gag	1500	24	**100.0** (2)	**100.0** (6)	99.33 (4)	**100.0** (3)	99.40 (10)
pol	3009	38	99.73 (4)	99.20 (3)	99.67 (2)	**100.0** (1)	98.44 (6)
vif	576	38	**100.0** (2)	98.96 (9)	**100.0** (3)	**100.0** (1)	97.40 (5)
nef	618	60	**100.0** (2)	97.14 (6)	97.30 (15)	97.62 (5)	96.02 (12)
env	2568	66	99.96* (1)	97.90 (11)	99.69 (7)	98.83 (12)	99.45 (25)

Table 4.6 Percentage sequence similarity of best recovered haplotype for each gene and HIV-1 strain pair. The bracketed figure indicates the rank of the best haplotype for the strain amongst all recovered haplotypes, according to its likelihood score. We also report the total number of haplotypes recovered by `Gretel` for each gene. *Recovery of 89.6 *env* gene has just one incorrect SNP (2567 SNPs recovered). †Number of haplotypes returned after conservative -1000 log_{10} likelihood cutoff.

Section 4.5

Experiment 4
Recovering haplotypes from a benchmark mock community

The previous *in silico* experiments have worked to show the potential of my haplotype recovery method. Finally, to show that `Hansel` and `Gretel` are capable of recovering haplotypes at scale, I applied the method to a synthetic metagenome. Such mock community benchmarks are necessary as there are currently no gold-standard annotated metagenomes on which to evaluate the method [216, 141].

This Section demonstrates:

- `Gretel` can scale to handle reads from a realistic microbial community
- `Gretel` can recover haplotypes without the need for pre-processing binning and filtering
- `Gretel` can recover haplotypes for hundreds of bacterial genes with high accuracy

4.5.1 Method

We apply `Hansel` and `Gretel` to a synthetic mock community introduced by Quince *et al.* (2017). The community contains 5 *Escherichia coli* strains, and 15 other genomes commonly found in the human gut according to samples from the Human Microbiome Project [244]. Reads were generated by the authors to simulate a "typical HiSeq 2500 run" [141]. The original work generated 1.504×10^9 reads, distributed across 64 paired-end samples (11.75M read pairs). As part of their preprint, the authors made available a subset of the mock community. The subset contains 16 samples of 1 million read pairs, totalling 32 million reads. The original paper also identified 982 single-copy core species genes (SCSGs) for *E. coli*. DNA sequences for the 982 genes, as found in the five different *E. coli* strains were provided by the authors.

As described by the flow diagram in Figure 4.14, the 16 read sets were co-assembled with `MEGAHIT` [222]. Contigs shorter than 1 kbp were discarded from the assembly as per protocol recommended by Quince *et al.*[1]. The resulting assembly is described in Table 4.7. The 982 provided SCSGs for each strain were mapped to the assembled pseudo-reference with `blastn`[2]. We required at least 75% of the gene's length to be mapped to the assembly to include the SCSG in our analysis. 814 of the 982 genes matched this criteria. In the scenario where `blastn` did not unanimously assign the five strains of a given SCSG to the same contig on the pseudo-reference, the most supported contig was selected.

The reads were mapped to the pseudo-reference with `bowtie2` (`--sensitive-local`). `Gretel` was then executed on the aligned reads, once for each of the 814 identified SCSG regions with the aim of recovering the five strain haplotypes from the synthetic short-reads. SNPs were called over each region using the `snpper` method previously described (Section 4.1.1). Performance was measured with a `blastn` alignment between the known five strain haplotypes, and the `Gretel` recovered haplotypes[3]. As for our synthetic evaluation, each input haplotype is assigned a best output haplotype, and an output haplotype may be the best haplotype for more than one input. For each strain, we report the sequence identity of the best haplotype for each of the 814 SCSG regions.

Method Summary

Existing Work

- Mock microbial community introduced by Quince *et al.* (2017)
- Community of 20 genomes including 5 *E. coli* strains
- 11.75M synthetic read pairs generated per sample from community, with 64 samples
- Authors made available 16 samples of mock community, with 1 million read pairs each
- 982 single-copy core species genes (SCSGs) for the 5 *E. coli* strains identified

Contributions

- 16 million read pairs assembled with `MEGAHIT`
- Reads aligned to pseudo-reference with `bowtie2`
- 814 of 982 SCSGs could be aligned back to our assembly with `BLAST`
- `Gretel` executed over the 814 mapped sites
- Haplotypes recovered by `Gretel` compared to provided SCSG sequences

	Contigs	Total bp	Min	Average	Max	N50	Time
Raw Assembly	17,066	67,189,963	200	3,937	689,365	53,290	4605 s
$\geq$ 1kbp	6,357	61,651,258	1,000	9,698	689,365	63,517	-

Table 4.7 Statistics for `MEGAHIT` assembled read data from `DESMAN` preprint

[1] github.com/chrisquince/DESMAN/blob/master/complete_example/README.md, commit `9045fe2`

[2] Default parameters with no cut-off were used as high-quality alignments of COGs to references were anticipated. Mean e-value of alignments: $2.10 \times e^{-12}$, max (worst) e-value: $1.02 \times e^{-8}$

[3] Default settings, no cut-off. Mean e-value: $1.32 \times e^{-62}$, max (worst) e-value $1.08 \times e^{-59}$

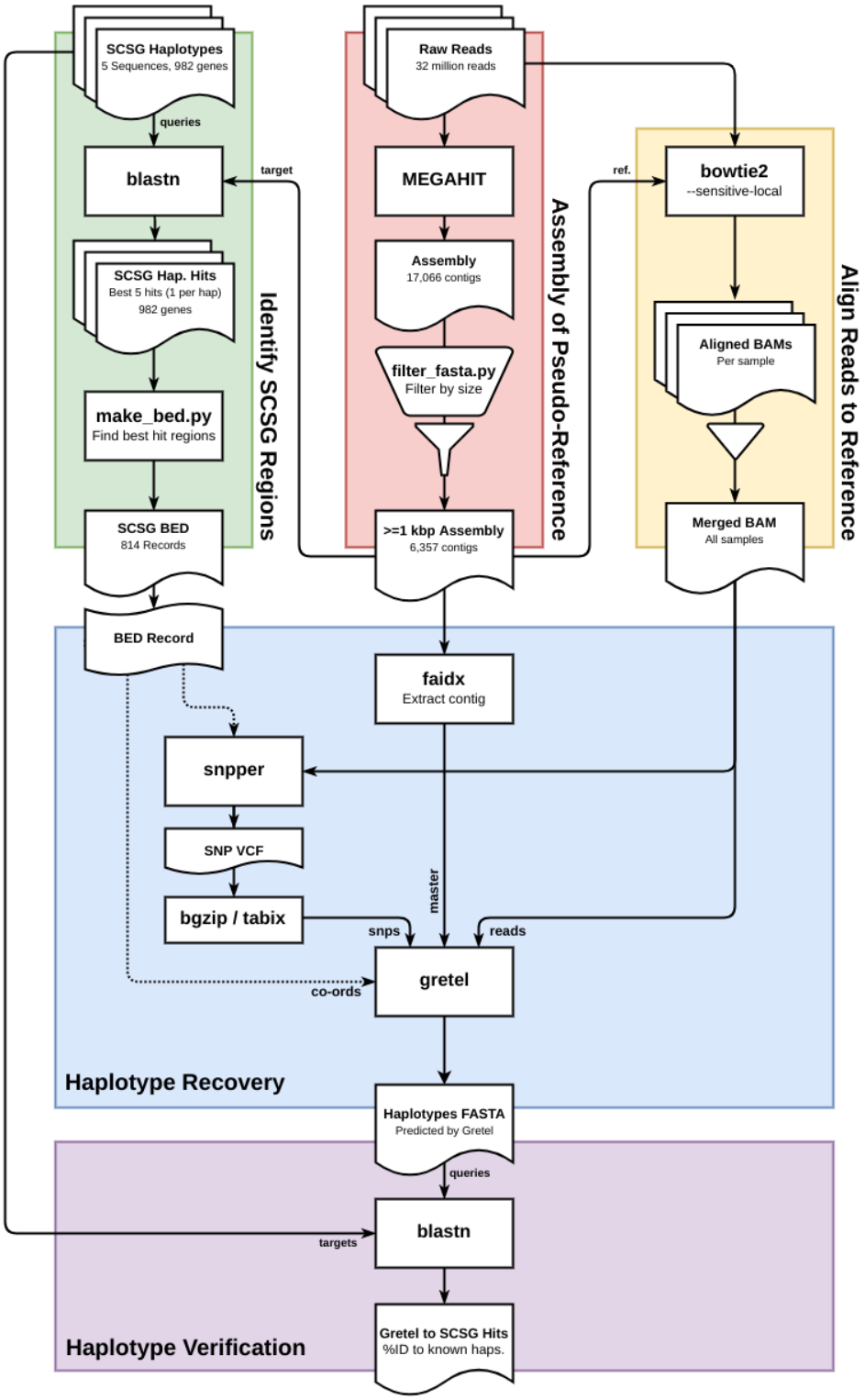


Figure 4.14 Flow diagram describing the computational pipeline for assembling and aligning raw reads, identifying single copy core species genes (SCSGs) on the assembly, and the recovery and validation of *E coli.* haplotypes across 814 SCSGs, from a mock microbial community.

4.5.2 Results

Table 4.8 presents summary statistics of the percentage identities of the best haplotype recovered by `Gretel` across the 814 genes, for each strain. Likewise, Figure 4.15 plots the distribution of percentage sequence identity (as measured by `blastn`) for the 814 `Gretel` recovered haplotypes to the original *E. coli* haplotypes.

We show `Gretel` is capable of achieving results with comparable accuracy to the current state-of-the-art for the related problem of strain de-convolution (`DESMAN` [141]). Yet, the binning step of the `DESMAN` pipeline led to a majority of the SCSGs being discarded, leaving only 372 (of 982) for their own analysis. Whereas `DESMAN` requires significant pre-processing, we have shown it is possible to achieve accurate haplotype recovery (over many more sites) without the need to perform any pre-processing. **Gretel is capable of scaling to recover strain-specific haplotypes from a microbial community, for hundreds of highly variable *E. coli* genes with an average accuracy of 99.52%.**

4.5.3 Conclusions

This experiment was performed to demonstrate that `Gretel` is capable of scaling to work with reads sampled from a community. In lieu of a real metagenomic dataset with known haplotypes, I used a *de facto* dataset provided by Quince *et al.* [141]. This mock community was also used as a means to compare my approach to that of `DESMAN`, as despite best efforts, I was unable to run `DESMAN` on my own synthetic data, representing the scenario of analyzing genes that are not SCSGs, with diversity present in a single microbial sample. As described earlier in my review (Section 2.8.4), `DESMAN`'s methodology is complex, and substantial effort is required to transform data into particular formats to fit through the different steps of their pipeline. This work shows that `Gretel` is competitive with the state-of-the-art in "strain identification", achieving comparable results over hundreds more regions of interest that `DESMAN` could achieve, without the need for pre-processing.

We observe that `Gretel` is particularly effective at recovering haplotypes from *E. coli K12*. It is unclear whether this is a bias in the abundance of *K12* in the synthetic reads, or the alignment of the SCSGs to the assembled contigs. It is difficult to unpick this, as the alignment of SCSGs to the assembly aimed to find a region supported by as many of the strains for each COG as possible, which could have been influenced by better alignments for *K12*. I am aware *K12* is more distantly related to the other four strains, than they are to each other [245], which could possibly account for the differences in recovery.

It is important to note that the construction of testing data to simulate metagenomes remains a research endeavour in itself [246, 247], and the Quince *et al.* community used here is one of the best methodologies available to test our approach. However, we[4] still wanted to empirically verify the approach with a **real metagenome**, requiring *in vitro* work, which leads me to my next chapter.

[4] Prompted somewhat by reviewer three...

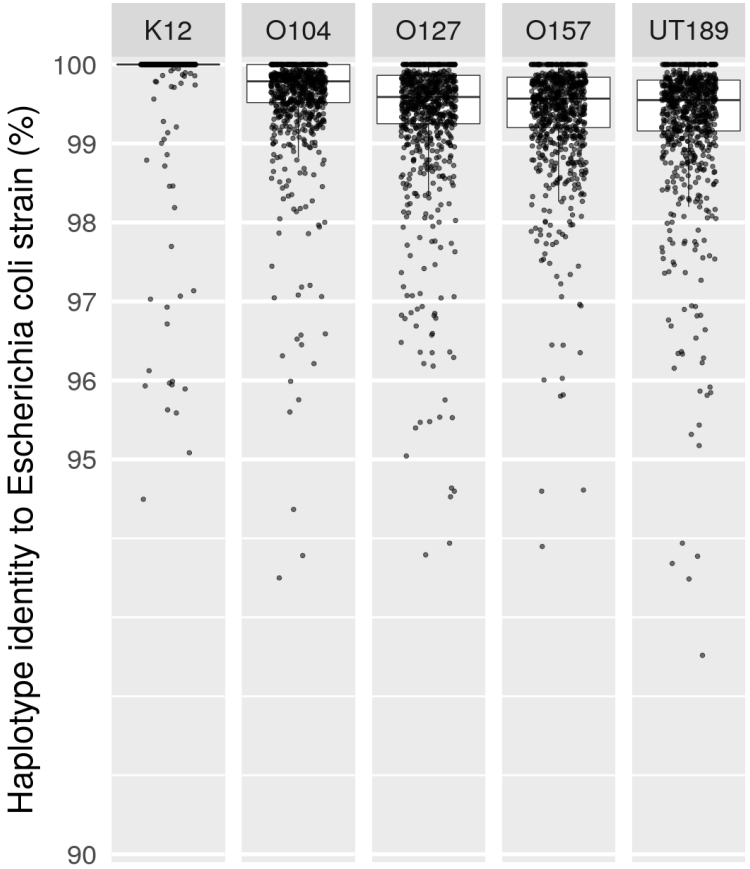


Figure 4.15 Boxplots summarising the percentage sequence identity (y-axis) of the best `Gretel` haplotypes recovered from each of the 814 gene sites, to five *E. coli* strains (column facets) known to exist in the mock community. `Gretel` was executed at 814 sites on an assembled mock metagenome, consisting of short-reads generated from five *E. coli* strain haplotypes, and 15 other genomes. The y-axis is truncated at 90%.

Strain	Min.	Mean	Median	Max.
K12	94.50	99.91	100.0	100.0
O104	93.50	99.61	99.79	100.0
O127	93.79	99.36	99.59	100.0
O157	93.90	99.40	99.57	100.0
UT189	92.52	99.31	99.55	100.0
All	92.52	99.52	99.75	100.0

Table 4.8 Minimum, mean, median and maximum percentage identity of the best haplotypes as recovered by `Gretel`, to the actual DNA sequences of the 814 genes, for each of the five *E. coli* strains.

Chapter 5

Recovery of enzyme haplotypes from the rumen microbiome

Finally, to validate `Gretel` empirically, I predicted haplotypes from short read sequencing of a natural microbial community, and verified their existence by sequencing isolated amplicons with long-read nanopore strand technology.

This chapter will describe:

- Identification of regions of biochemical interest from a rumen metagenome
- `Gretel` recovery of haplotypes from sequenced DNA that overlap these target regions
- Haplotype-directed design of primers and amplification of sequences from RNA
- Sanger and Nanopore sequencing of amplicons
- Validation of recovered haplotypes by comparison to long-read sequences

I present the first biologically validated method for the recovery of haplotypes from a real microbial community.

5.1 Methodology

5.1.1 Existing Data

Huws *et al.* [104] isolated RNA from 30 rumen samples from 3 cows over 5 timepoints (1, 2, 4, 6 and 8 hours after feeding), with two replicates (Table 5.1). In preparation for metatranscriptomic sequencing, the polyA fraction was removed (MicroPoly(A)Purist, Ambion). 18S rRNA was also removed (both RiboMinus Plant Kit and Eukaryote Kit, Invitrogen). 16S rRNA was removed (Ribo-Zero bacterial rRNA removal kit, Epicentre) all according to the manufacturer's protocols. The resulting enriched microbial mRNA was prepared for sequencing using TruSeq Stranded mRNA Library Prep kit (Illumina). Subsequently, the library was sequenced using an Illumina HiSeq 2500 (2 $\times$ 100 bp, Section A.2.2). 118 million paired-end reads were generated and are deposited under the ENA study `PRJNA419191`.

As part of a follow-up work whose manuscript is in preparation at this time, a previous PhD student – Francesco Rubino – partitioned the reads with `khmer` [248] and assembled with `Velvet` [231]. The assembly was enriched with decoys including a draft copy of the Hungate genomes [136] and several plant genomes. Proteins were predicted and annotated with Enzyme Commission (EC) numbers to produce a GFF file with `MGKit` [123] with the `Uniprot` database [249].

5.1.2 Filtering candidate regions

Each sample's original archived short sequence reads were re-aligned to the existing assembly with `bowtie2` (`--local`) before merging all samples with `samtools merge` to create one canonical alignment of all reads (248,092,426 alignments). To recover industrially relevant enzyme isoforms from the metatranscriptome, we focused our attention on hydrolases known to be found in the rumen [105]:

- **EC 3.2.–** glycosylases
- **EC 3.4.–** peptidases
- **EC 5.3.–** intramolecular oxidoreductases (isomerases)

	1 h	2 h	4 h	6 h	8 h
Cow 1	3, 4	9, 10	15, 16	21, 22	27, 28
Cow 2	5, 6	11, 12	17, 18	23, 24	29, 30
Cow 3	7, 8	13, 14	19, 20	25, 26	31, 32

Table 5.1 Sample numbering manifest for each animal, timepoint and replicate.

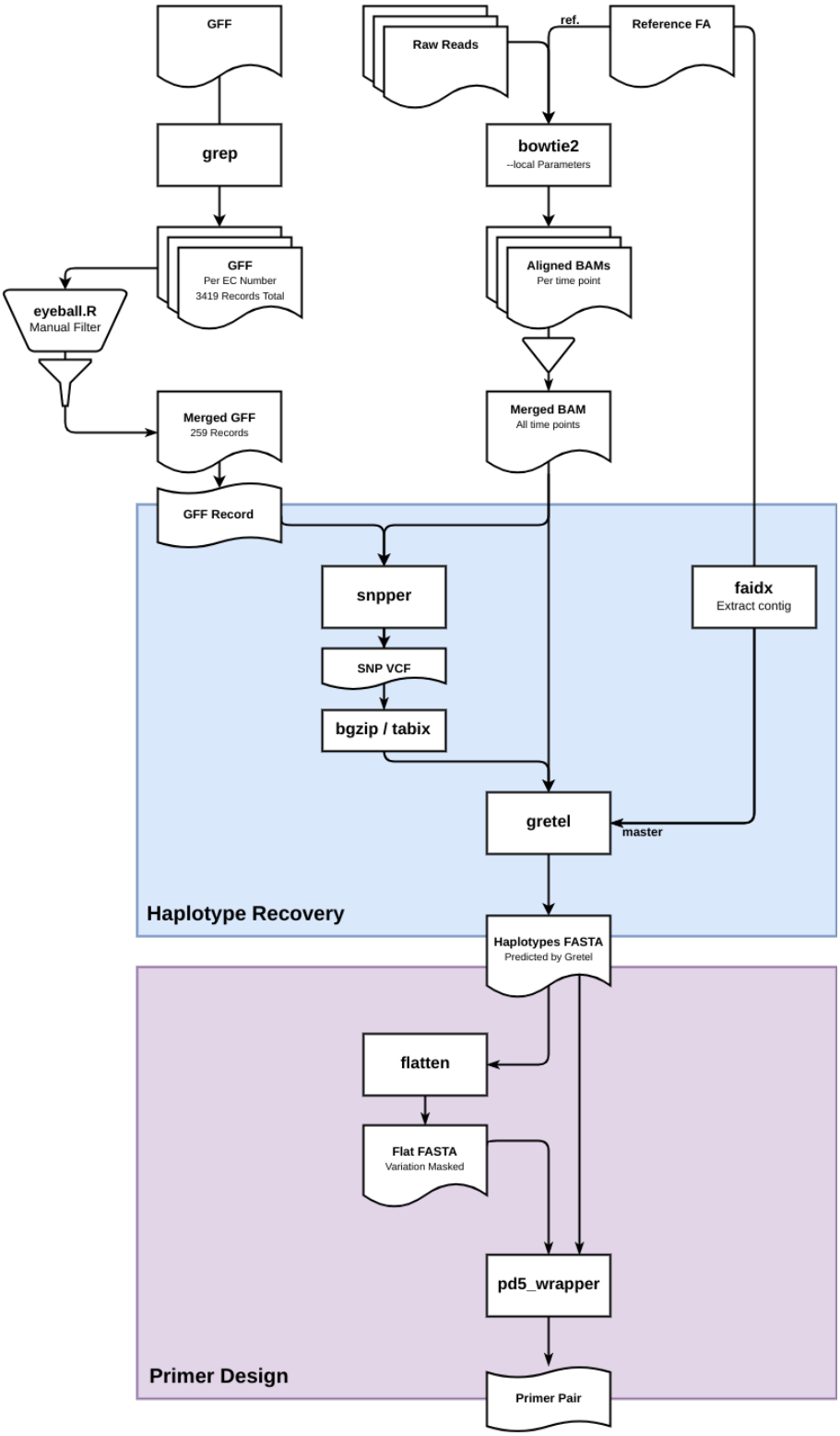


Figure 5.1 Flow diagram depicting the computational pipeline used to select regions of interest from the GFF, align reads to the reference, recover corresponding haplotypes with `Gretel` and generate suitable primers.

The existing GFF annotations were filtered to create a subset of all entries with Enzyme Commission (EC) numbers 3.2, 3.4 or 5.3. 3,419 regions from the GFF were identified and were cross-referenced to the new read alignment. After eyeballing summaries of the data in `R`, regions were filtered with the following criteria:

- minimum coverage $\geq$ mean minimum coverage (19.7x)
- length $\geq$ new mean region length (615.7)
- standard deviation of coverage $\leq$ average standard deviation of coverage over remaining regions (76.79x)

Filtering returned 259 possible candidate regions. `Gretel` was individually executed over the 259 regions; recovering a total of 7536 haplotypes from the merged aligned reads. Figure 5.1 depicts the computational pipeline used to find regions of interest, and generate primer pairs.

5.1.3 Final selection and primer pair design

With a view to amplify a subset of the 259 candidate regions from cDNA for sequencing, we need to design primers. To be able to actually verify haplotypes, we additionally need our primers to retain the ability to bind to as many of a region's recovered haplotypes as possible. For a particular candidate region, we first consider each of the haplotypes in order of descending likelihood and create a corresponding "flattened" consensus by flipping any base that disagreed with the base call of any haplotype with a better likelihood, to an 'N'. That is, we iteratively built a consensus by applying the next best haplotype onto the current consensus and masking out any variation (Figure 5.2).

A valid primer pair for a given haplotype's consensus sequence should bind to that haplotype *and* all haplotypes with a better likelihood. For each region, our goal was to identify a primer pair with the longest template length, on a consensus that represented as many haplotypes as possible.

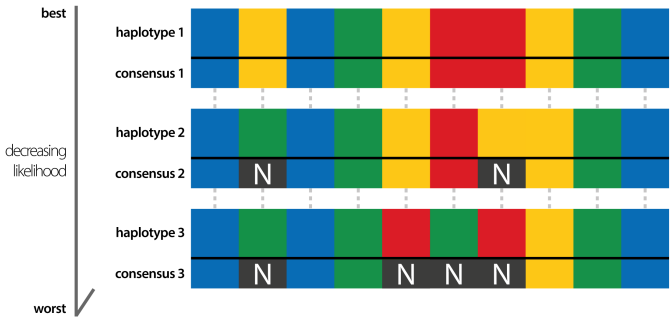


Figure 5.2 Depiction of haplotype consensus generation. Haplotypes (coloured sequences) are ordered by decreasing likelihood. Bases on a haplotype where any previous (higher likelihood) haplotype do not share a consensus are flipped to an 'N'. The goal was to find a pair of forward and reverse primers that could maximise the template length of a gene, and also cover as many of the recovered haplotypes as possible.

pd5[250] was executed on each consensus to find an appropriate forward and reverse primer. Primers could be between 25 and 40 nt, with a calculated annealing temperature between 55 and 65°C. The minimum target region was considered to be after the first and before the last 100 bp of the haplotype. That is, a valid forward primer must end anywhere within the first 100 bp (and conversely for the reverse). For laboratory analysis, 10 regions were hand-selected, considering the criteria:

- gene length
- primer template length
- number of predicted haplotypes
- distribution of haplotype likelihoods
- evidence of similar gene sequence in databases
- number of haplotypes that could be captured by generated primers

For each of the 10 chosen regions, a corresponding pair of *ThermoFisher Custom Value Oligos* were synthesized (desalted, 25 nmol scale). Table 5.7 describes the regions selected for further analysis with their associated oligo sequences.

5.1.4 Reverse Transcription and PCR

Stock RNA from the 30 samples was pooled[1] according to the density of reads that mapped to the selected regions by timepoint (Figure 5.3). 13 gene-specific reverse transcription reactions were performed with the *QuantiTect Reverse Transcription Kit*[2], according to manufacturer's protocol. Each of the 10 selected regions had an individual corresponding cDNA library, along with two positive controls (random hex-mer as included with the kit, and a common rRNA primer) and one negative control (no primer). PCR was performed for each of the 10 genes, using the corresponding cDNA and primer pair. Figure 5.7 illustrates the workflow for the reverse transcription and PCR reactions.

Qiagen QuantiNova vs. QuantiTect
It is important to note for future work that the *Qiagen QuantiNova* kit incorporates random hex-mer primers in its buffer. An initial attempt to transcribe and amplify our samples with the *QuantiNova* kit yielded poor results, most likely due to an overamplification of ribosomal RNA providing too many alternative non-specific binding sites for our primers to find their targets. Using the alternative *Qiagen QuantiTect* kit – which packages the random primers separately, allowing them to be omitted from the master mix – permits gene-specific reverse transcription, and yielded PCR products from the samples as expected.

[1] Given an approximate RNA concentration of $0.5\,\mu g\,\mu l^{-1}$, 26 µl of stock was made up of: 2 µl from t_{1hr} and t_{2hr}, and 11 µl from t_{6hr} and t_{8hr}, to provide 1 µg of RNA template for each of the 13 reverse transcription reactions.
[2] Qiagen Cat. 205311

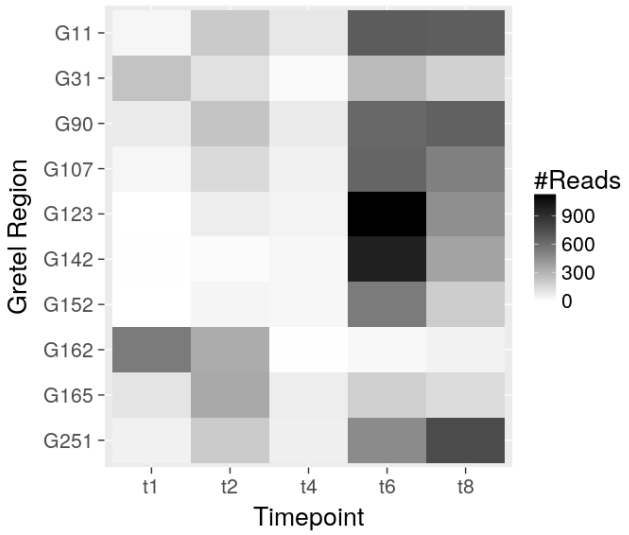


Figure 5.3 Heatmap showing the number of reads aligning to the 10 selected regions, by timepoint after the introduction of feed to the animal rumen. Read density was used to select which RNA sample tubes to pool before reverse transcription. 2 µl of samples from t_{1hr} and t_{2hr} and 11 µl from t_{6hr} and t_{8hr} were selected.

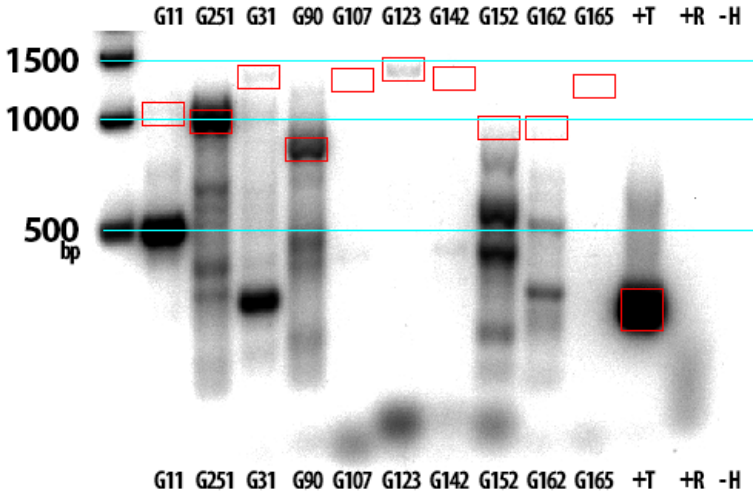


Figure 5.4 First gel electrophoresis result from amplification of the ten chosen Gretel candidates. Approximate expected lengths are marked by red boxes. Several candidates were successfully amplified at the correct length and were selected for further analysis (see gradient below).

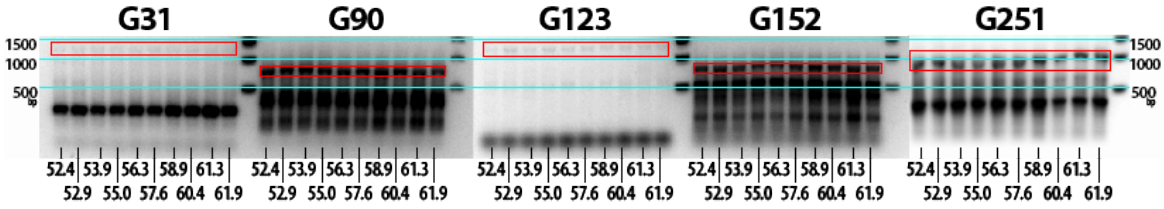


Figure 5.5 Gel electrophoresis result after conducting PCR with an annealing temperature gradient between 52°C and 62°C, for the five candidates selected for further analysis. Desired product lengths marked in red.

Initial exploratory work was performed with *Promega GoTaq G2 Polymerase*, according to the manufacturer's instructions (see Appendix B.1 for reaction mix and thermocycler programme). Results were confirmed via electrophoresis (0.5% agarose), indicating that several candidates had been successfully amplified at the expected lengths (Figure 5.4). Five candidates (G31, G90, G123, G152 and G251) were selected for further investigation.

In an attempt to optimise the PCR reactions for the selected candidates, a 52°C – 62°C annealing gradient was performed (with the same reagents and parameters). The gradient yielded homogeneous banding (Figure 5.5), indicating a robust reaction with a stable annealing temperature.

5.2 Results

5.2.1 Sanger Sequencing

Sanger sequencing (Section A.2.1) was employed to validate whether the amplified sequences had identity to the target genes. The method uses chain-terminating dideoxynucleotides to reconstruct fragments that are terminated by bases attached to fluorescent labels. The fragment's length and its terminating label tells us what base is at the position that corresponds to the fragment size. The sequences are determined via capillary electrophoresis, which pulls the fragments in size order, past a laser that can read the fluorescent labels, yielding a *chromatogram* that can be interpreted to determine the nucleotide at each position of the sequence.

The process yields high-quality base calls, and substantially longer reads compared to typical shotgun sequencing methods. However, the start and and of the reads are typically unreliable, and still cannot cover a whole gene (with reads between 500 - 700 bp). Although the chromatogram can show whether a given position may have variants, it is not possible to determine which variants co-occur with one another, or whether they could be errors. To be clear, Sanger sequencing is not suitable for haplotyping, but provides us with an initial high-quality consensus of the amplicons, with limited insight into their variation.

150 µl of each of the five candidates (G31, G90, G123, G152 and G251) remained after the gradients were run, which was pooled and separated via gel electrophoresis. DNA was isolated from excised bands using the *Qiagen QIAquick Gel Extraction Kit* (with minor modification; Appendix B.2) and prepared for Sanger sequencing (Appendix B.3), which was performed at the Translational Genomics Facility, Aberystwyth.

Sample	Fwd	Rev	blastn	blastx
G 11-170706-A	470	461	*Uncultured rumen bacterium clone*	hypothetical protein
G251-170706-A	1081	858	*Butyrivibrio fibrisolvens*	glucose-6-phosphate isomerase
G 90-170706-A	453	426	*Uncultured rumen bacterium clone*	hypothetical protein
G 31-170713-A	–	–	–	–
G 90-170713-A	782	787	*Prevotella ruminicola*	alpha-N-arabinofuranosidase
G123-170713-A	1306	–	*Butyrivibrio proteoclasticus*	endo-1,4-beta-xylanase
G152-170713-A	637	836	*Prevotella melaninogenica*	30S ribosomal protein
G251-170713-A	1008	1033	*Butyrivibrio fibrisolvens*	glucose-6-phosphate isomerase
G 31-170726-A	842	631	*Fibrobacter succinogenes*	glycoside hydrolase
G123-170726-A	–	233	–	–

Table 5.2 Summary of Sanger sequencing results for selected `Gretel` candidates. Each row reports the length (bp) of the forward and reverse sequencing run for a given candidate, isolated from a gel after PCR and electrophoresis, and the top `blastn` and `blastx` hit for the sequence using the NCBI BLAST API (default parameters). For G31, G90, G123, G152 and G251, at least one sample could be found in NCBI databases.

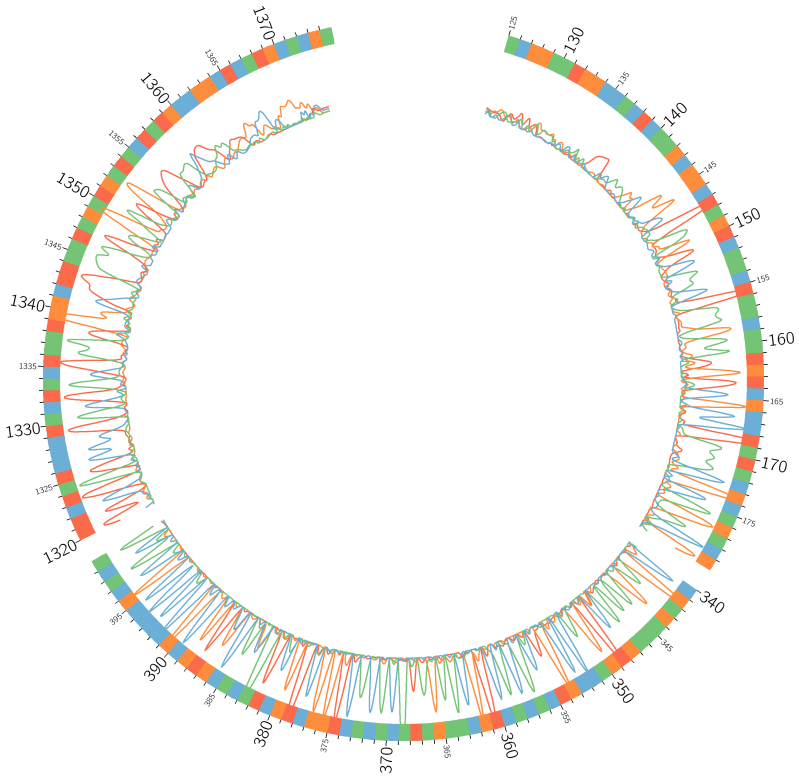


Figure 5.6 Partial chromatogram plotting Sanger sequencing signals for sample `G123-170713-A`. The outer ring represents the original reference sequence and position. The four coloured lines represent the signal observed for each of the four different nucleotides across each position of the sequenced DNA template (moving clockwise from the start, and counter clockwise from the end – recall Sanger reactions are performed in the forward and reverse directions). Note the lower quality signal at the start and end (reversed start) of the sequenced data. The "middle" partial chromatogram demonstrates clear signals, with defined peaks occupying each single position of the reference. We can observe SNPs as multiple peaks (*e.g.* 347, 355, 386), but sequencing alone cannot inform us as to how they are linked together by haplotypes.

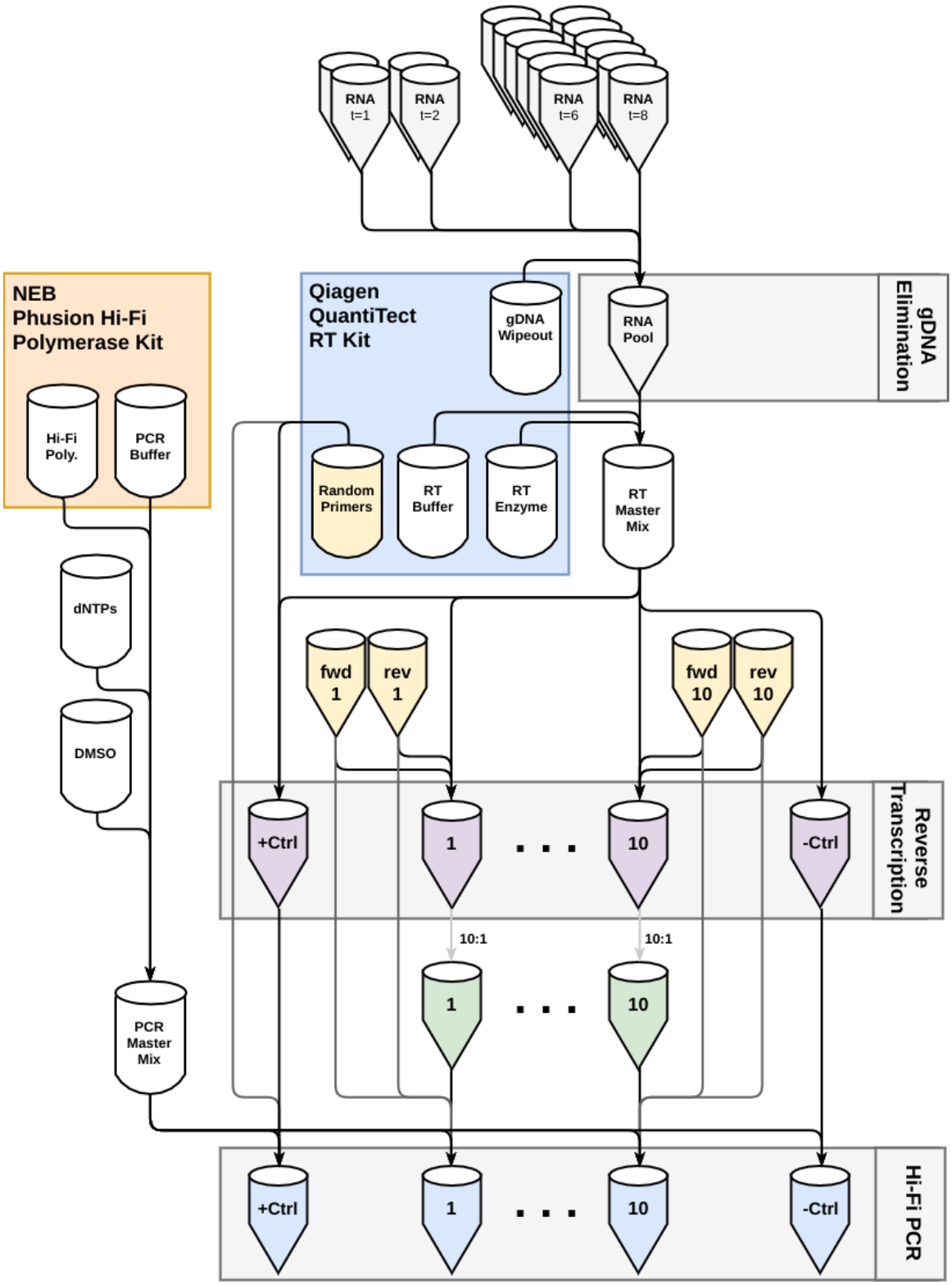


Figure 5.7 Flow diagram depicting the protocol for gene-specific reverse transcription and high-fidelity PCR

5.2.2 High-Fidelity Amplicons

For in-PCR amplicon error correction, the original 13-tube (10 genes, 2 positive, 1 negative) experiment was repeated with *NEB Phusion High-Fidelity DNA Polymerase*[3] (Figure 5.7). PCR was conducted according to the manufacturer's instructions (with 30 cycles, 65°C annealing, 60s extension; Appendix B.4) with each of the 10 genes' corresponding cDNA (in a 1:10 dilution to conserve stock) and primer pair. Although the high-fidelity polymerase yielded clearer bands (Figure 5.8), the reaction was less robust and amplifying candidates with sufficient mass proved difficult, even for template that had worked previously. Unable to amplify G152, G11 was selected to take its place, despite bands indicating the product was the wrong size, we were curious to explore its strong signal.

Annealing temperature for NEB Phusion
It is of note for future work that the *NEB Phusion* polymerases recommend a significantly higher annealing temperature than its competitors. An initial attempt to use *NEB Phusion* yielded no bands where there had been previously with the *Promega GoTaq*. Increasing the annealing temperature from 58°C to 65°C provided significant improvement (Figure 5.8).

Recovered DNA was used as the template for two parallel runs of high-fidelity PCR (Figure 5.9) to generate sufficient molecular weights for the Nanopore protocol, which recommends 1500 ng of product. Bands were excised following gel electrophoresis, and DNA extracted using the *Qiagen QIAquick Gel Extraction Kit* (with minor modification; Appendix B.2). Sufficient quantities of DNA were taken from each of the ten resulting samples to provide roughly equal molarity of each of the five targets when the samples were pooled. G31, G90, G123 and G251 were selected for further analysis, as they could be produced at the expected length and in adequate amount for Nanopore sequencing. We would later discover G11 was contaminated with rRNA carryover from the reverse transcription.

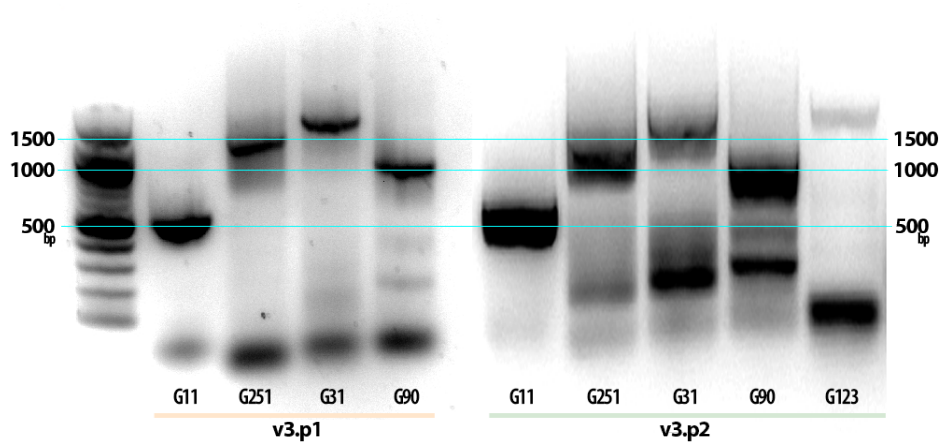


Figure 5.8 Gel electrophoresis banding to confirm high-fidelity PCR runs

[3]NEB Cat. M0530S

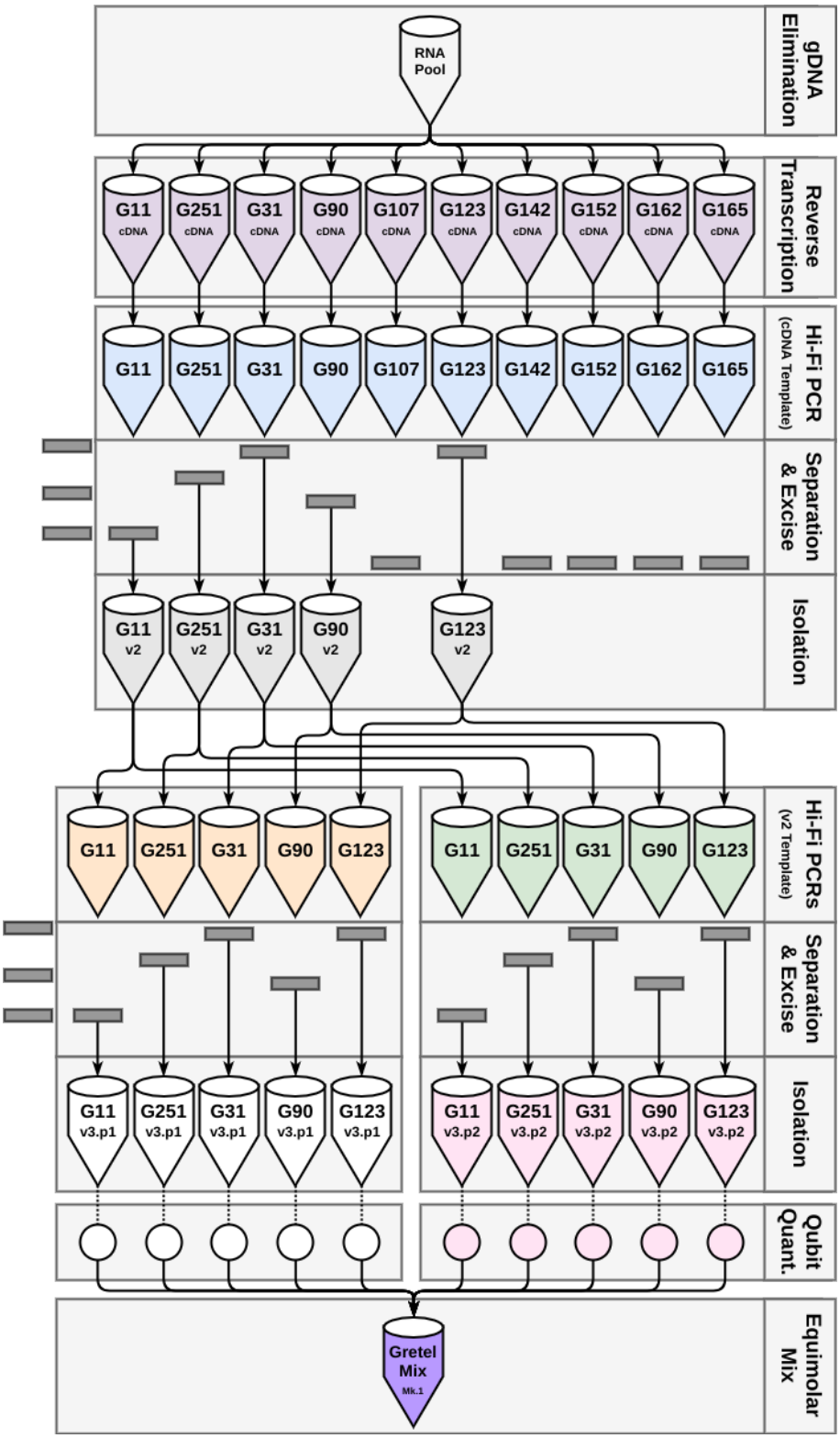


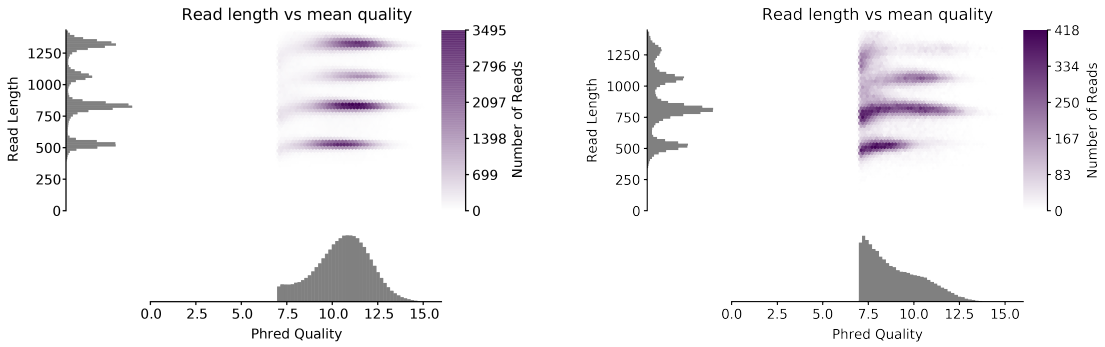
Figure 5.9 Flow diagram depicting the pipeline to generate amplicons for Nanopore sequencing

5.3 Oxford Nanopore sequencing

To verify our haplotypes, we needed to sequence our amplicons with a platform that would not require the reads to be assembled. Recall that nanopore strand sequencing pulls single molecules of DNA through a pore the width of a human hair, generating base calls from the observed changes in voltage caused by the molecule passing through the protein pore (Section A.2.4). Although the technology yields long reads, it currently requires substantial input DNA (necessitating multiple runs of PCR, electrophoresis and extraction) and has a relatively high error rate compared to current short-read sequencing technologies (Section A.2.2).

Amplicons were pooled in a ratio that attempted to equalize the molarity of the ten inputs (two sets of five amplicons) in the recommended 1500 ng. Molarity was estimated via assay on a *Qubit 2.0 Fluorometer*. The pooled DNA volume was 433.8 µl and required concentrating. DNA was recovered by following an *AMPure Bead Cleanup* protocol (60% bead concentration) and resuspended in 46 µl nuclease-free water. We followed the *Oxford Nanopore SQK-LSK108* laboratory protocol to prepare a library for sequencing (Appendix B.5.2). For future work, I set aside 500 ng of recovered product after the first ethanol wash, and completed the protocol with 354 ng of DNA amplicon. Prepared DNA was loaded onto a *FLO-MIN106* flowcell. The platform test returned 1,402 viable single cell pores. Sequencing was performed with `MinKNOW` (v1.7.14) running an unmodified `NC_48Hr_sequencing_FLO-MIN106_LSK108` protocol.

The run was manually terminated after 1h 28m 35s and yielded 672,388 reads. Base calling was completed with `Albacore` (v2.02). 634,859 reads passed quality control, and are available via ENA study `PRJEB23483`. Figure 5.10a shows the distribution of `phred` quality scores across reads. The mean score of 10.53 corresponds to an error rate of 8.85%. Despite this, we were able to identify individual molecules with extremely high identity to recovered haplotypes.



(a) All QC passed reads

(b) 111,697 remaining unaligned reads

Figure 5.10 Joint distributions of read length (y-histograms) and corresponding `phred` quality scores (x-histograms) across the MinION sequenced reads: (a) after `Albacore` basecalling and automatic quality filter, and (b) for the remaining reads that could not be aligned (Section 5.4.1). Intensity of purple heatmap depicts the number of generated reads of a given length and quality. Plots generated by `pauvre` [251].

Method Summary

Existing Work

- 30 RNA samples, from multiple timepoints and cows, sequenced on Illumina HiSeq
- Reads assembled with `Velvet`, enriched with Hungate genomes
- Assembly annotated with *Swiss-Prot* reviewed proteins by `MGKit`

Contributions *in silico*

- Annotations filtered by EC numbers 3.2, 3.4 and 5.3
- Filtered annotations manually curated, yielding 259 candidate regions
- `Gretel` executed over 259 regions to recover haplotypes
- Primers designed with `pd5`, directed by haplotypes
- 10 regions selected for *in vitro* analysis based on primer efficacy

Contributions *in vitro*

- Gene-specific reverse transcription of archived RNA to make cDNA
- PCR amplification of 10 cDNA libraries corresponding to selection regions
- Isolated amplicons submitted for Sanger sequencing
- Two rounds of Hi-Fi PCR to generate sufficient mass for Nanopore
- Nanopore sequencing of five candidate amplicons

5.4 Haplotype Verification

5.4.1 Post-sequencing analysis

Nanopore reads were aligned to the original 259 corresponding regions of the pseudo-reference with `minimap2`[4] [252] (`-ax map-ont --secondary=no`). The results are summarised in Table 5.3. We show that four of the five references are correctly amplified from the cDNA, with thousands of reads aligning to each region. However, G11 was not amplified at all. Some reads are aligned off-target (non-bold entries in table). A brief investigation revealed that (with the exception of G150), the off-target regions all correspond to *glucose-6-phosphate isomerases*, the target protein for candidate G251. Generally, the off-target regions tend to have a smaller breadth of coverage, and higher variation in coverage depth. Note that we do not expect 100% coverage breadth even in the best case, as the generated primer pairs were permitted to lie within the first and last 100 bp of the targeted regions.

Of the 634,859 reads that passed quality control (94.42%), 374,356 (58.97%) could be aligned to one of the references, leaving 260,503 (41.03%) unmapped.

Region by Number	Reads Mapped	Region by Reads	Reads Mapped	Region Covered (%)	Scaled IQR Coverage (×)
G11	0	**G90**	161,740	92.07	0.032
G31	77,421	**G123**	101,426	94.74	0.030
G90	162,726	**G31**	77,089	93.92	0.035
G123	101,724	**G251**	15,293	90.81	0.030
G150	1	G254	8,837	71.61	0.014
G227	2,950	G255	2,787	64.75	0.070
G232	2,196	G227	2,678	94.03	0.107
G237	228	G232	2,158	72.23	0.212
G238	313	G250	1,799	94.67	0.252
G249	223	G238	288	91.82	0.131
G250	1,891	G237	160	43.80	0.031
G251	15,515	G249	100	63.71	0.055
G254	8,946	G150	1	38.51	0.0
G255	3,250	**G11**	0	0.0	–

Table 5.3 Number of Nanopore reads mapped back to the 259 candidate regions on the pseudo-reference. The regions corresponding to the five chosen candidates are highlighted in bold, and regions with 0 reads are not shown. Regions are ordered by their identifier, and alternatively, number of reads descending. Note that we failed to amplify G11 (the amplicons had yielded rRNA contaminant), and the presence of off-target alignments (non-bold) correspond to additional *glucose-6-phosphate isomerases* regions with identity to G251. Coverage percentage refers to proportion of positions across the region with at least one read included in a read pileup. IQR refers to the interquartile range of coverage for all such positions, after scaling coverage between 0 and 1.

[4] `minimap2` has become the *de facto* read aligner for long read sequencing, and addresses problems with `bwa-mem`'s performance on such reads. Heng Li's work on `minimap2` has been accepted for publication in *Bioinformatics* but its release has been delayed: https://lh3.github.io/2018/04/02/minimap2-and-the-future-of-bwa

Aligning the unmapped reads to the SILVA small subunit rRNA database [115], can account for 107,145 (41.13%) of the unmapped reads, indicating rRNA contamination, most likely from the failed attempt to amplify candidate G11. Table 5.4 outlines the most common assignments (using the headers from the database), with almost all entries falling under the *Lachnospiraceae* family. *Lachnospiraceae* are a family of clostridia that describe strictly anaerobic bacteria, commonly found in the in gut of animals (particularly ruminants), with the ability to degrade pectin [253], which is consistent with what one would expect to find in the original cDNA.

Reads (Num.)	**Reads (%)**	**SILVA Entry Sequence Description**
58,148	22.32	Lachnospiraceae ▶ Butyrivibrio 2 ▶ uncultured (*) bacterium
17,036	6.54	Lachnospiraceae ▶ Butyrivibrio 2 ▶ Butyrivibrio hungatei
6,600	2.53	Lachnospiraceae ▶ Butyrivibrio 2 ▶ *
3,697	1.42	Lachnospiraceae ▶ Butyrivibrio 2 ▶ *Butyrivibrio fibrisolvens*
3,037	1.17	Lachnospiraceae ▶ Butyrivibrio 2 ▶ *Butyrivibrio sp. FCS014*
2,797	1.07	Lachnospiraceae ▶ Lachnospiraceae NK3A20 group ▶ *
2,405	0.92	Lachnospiraceae ▶ Butyrivibrio 2 ▶ rumen bacterium R-26
1,828	0.70	Lachnospiraceae ▶ Lachnospiraceae NK4A136 group ▶ uncultured bacterium
1,035	0.40	Lachnospiraceae ▶ Lachnoclostridium ▶ *
816	0.31	Lachnospiraceae ▶ Roseburia ▶ *
8,981	3.45	Other Lachnospiraceae
416	0.16	Other Ruminococcaceae
349	0.13	Other Bacteria
153,538	58.87	Unmapped

Table 5.4 Raw count and proportion of unmapped reads (n=260,503) and their strongest `blastn` hit to an entry in the SILVA small subunit rRNA 'SSU Ref NR 99' database (Version 132). Hits were filtered to have a bitscore of at least 100. 'Other' refers to aggregated groups of taxons with fewer than 1000 hits.

Reads (Num.)	**Reads (%)**	**GenBank Accession Sequence Name**	**Uniprot Accession Top `blastx` Hit**
26,199	17.03	`FP929036.1` *Butyrivibrio fibrisolvens* 16/4 draft genome	`D4IXB5` *glucose-6-phosphate isomerase*
3,657	2.38	`CP002006.1` *Prevotella ruminicola* 23, complete genome	`D5EX25` *alpha-N-arabinofuranosidase*
1,592	1.04	Other Mapped (only one read, n=1592)	–
7,480	4.87	Other Mapped (more than one read, n=562)	–
114,430	74.62	Unmapped	–

Table 5.5 Remaining unmapped sequences (n=153,358) were aligned with `blastn` against `nt`. An additional 38,928 reads (25.39%) could be explained, and feature some identity to the candidate regions of interest. `blastn` results were filtered to have a bitscore of at least 100. 'Other' consists of all accession whose proportion of assigned reads did not reach 1.0%, and are subdivided by those with a single aligned read (1,592) and the 562 other accessions with more than one read (14.36 read hits on average).

Of the remaining 153,358 reads that did not map to either one of the 259 candidate regions, or the SILVA SSU database, it was possible to align a further 38,928 (25.39%) reads to `nt` with `blastn`. Table 5.5 summarises the result. Of note are two regions on draft genomes for *Butyrivibrio fibrisolvens* and *Prevotella ruminicola* that are hit by 17% and 2.4% of the remaining reads, respectively. Additionally, the subsequences of the draft genomes against which the majority of these newly mapped reads align both correspond to an entry in Uniprot that matches the function of one of the 10 chosen candidates (G251 and G90, respectively). The results indicate a small subset of the previously unmapped reads actually have some affinity to the target sequences, but lack sufficient identity for `minimap2` or `bowtie2` to align them to one of the 259 original sequences. This intriguing result will be further investigated following this thesis. The highlighted organisms are not only consistent with what one would expect to find in the original samples, but the regions have affinity to those targeted for haplotype recovery.

Aligning the remaining 114,430 reads to the original assembly with `bowtie2` (`--sensitive-local --score-min L,60,0.6`)[5] accounts for a fraction of the reads that could not be mapped to one of the 259 references, the `SILVA` database, or `nt`. Table 5.6 describes the two contigs that account for 94.3% of the 2,733 reads that could be mapped to the assembly. Oddly, it was not possible to find a binding site for any of the 10 primer pairs upstream or downstream of either of the contigs presented in the table, so it is not exactly clear why the reads are binding at these positions.

Figure 5.10b shows the distribution of `phred` quality scores across the remaining unmapped reads. Note the distribution has shifted towards the lower `phred` quality scores, with a mean score of 8.99 (12.62% error rate). Lacking further evidence to the contrary, it is assumed that the 111,697 remaining long-read sequences exhibit too much noise or error to be meaningfully identified.

Reads (Num.)	**Reads (%)**	**Contig Name**	**Top `blastx` Hit**
2,199	1.92	*Butyrivibrio_sp._AE2015\scaffold00002*	*extracellular solute-binding protein*
378	0.33	*part2-group0048\NODE_3082*	*hypothetical protein*
12	0.01	Other Mapped (only one read, n=12)	–
144	0.13	Other Mapped (more than one read, n=12)	–
111,697	97.61	Unmapped	–

Table 5.6 Remaining unmapped sequences (n=114,430) were aligned with `bowtie2` against the original reference assembly. An additional 2,733 reads (2.39%) could be explained. 'Other' consists of all contigs where the number of assigned reads was less than 100, and are subdivded by those with a single aligned read (n=12) and the other contigs with more than one read (n=12, 12 reads on average).

[5] Here, I adjusted the minimum scoring function to limit alignments to those that match at least $\approx 60\%$ of a read to avoid significant soft clipping of the long-reads causing spurious short alignments.

5.4.2 Haplotype Verification

Nanopore reads that had passed quality control were sensitively aligned to the five references corresponding to the chosen regions with `bowtie2` (`--sensitive-local`). A script (`hamming_reads.py`) was used to parse the `CIGAR` strings of the alignment, and calculate the Hamming distance of all reads against all recovered haplotypes for each gene. Due to the abundance of homopolymer runs and slippage in the sequenced Nanopore reads, I chose to ignore indels for the calculation of Hamming distance. The distance between each long-read versus each `Gretel` predicted haplotype was calculated for each of the five genes.

Figure 5.11 depicts[6], for G123, a comparison between several of the highest likelihood `Gretel` recovered haplotypes, and their associated highest identity sequenced DNA molecules. The outermost ring displays the original assembled pseudo-reference sequence, followed by a visualisation of the Illumina read coverage (light grey banding), and the Sanger sequencing chromatogram (Section A.2.1). Moving towards the centre of the diagram, pairs of bands compare single molecule sequences to predicted haplotypes. Haplotypes are masked (black) over sites that were observed to be homozygous by my naive SNP caller, `snpper` (Section 4.1.1). The diagram is "exploded" to focus on positions that demonstrate variation. Figures for the remaining genes can be found in Appendix C.

We can compare the colour (representing a particular base) at unmasked haplotype positions to that of the reads, and can determine that the sequenced reads support our haplotypes. Indeed, the haplotype with the best likelihood for Gretel G123 had a Hamming distance of 0.003 (representing sequence identity of 99.7%). Note that not only can putative SNPs can be observed on the chromatogram (where multiple coloured peaks support the existence of more than one base in the sample), but these peaks also align to positions where the haplotypes were predicted to have SNPs, and the coloured peaks support the bases on the recovered haplotypes.

5.5 Conclusion

We show that `Gretel` has predicted novel isoforms of an exoglucanase enzyme, with potential biotechnical applications.

I can now conclude, with *in vitro* evidence, that `Gretel` is capable of recovering haplotypes from a natural microbial community. For the first time we have shown that a computational method is capable of recovering sequences of co-occurring variants (haplotypes) that actually exist in nature, with high accuracy, from short-read data.

[6]The most beautiful graph I have ever made. I want this `circos` [254] plot on my wall.

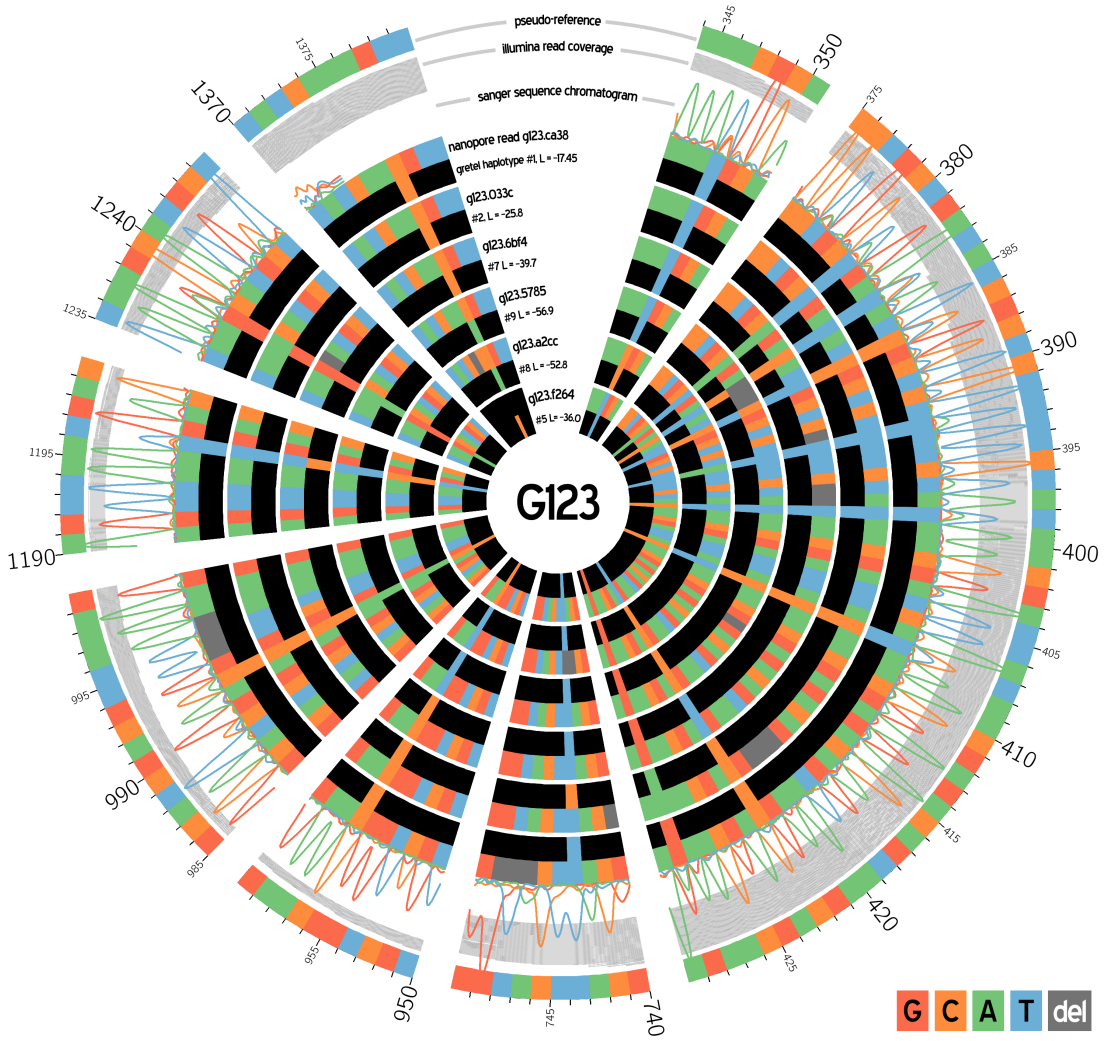


Figure 5.11 Comparison of our recovered haplotypes against Oxford Nanopore long-read data for Gretel G123 (*Exoglucanase XynX*). Outermost ring represents the metagenomic assembly (pseudo-reference). Grey banding represents coverage of original Illumina read data. Line plot depicts Sanger sequencing chromatogram for G123 PCR amplicon. Pairs of tracks toward the center align a DNA molecule sequenced by Oxford Nanopore MinION (outer, coloured) to a specific haplotype recovered by `Gretel` (inner). The haplotypes are masked (black) at sites homozygous over the displayed haplotypes to ease comparison of predicted variants. Heterozygous sites on the haplotypes are supported by Sanger sequencing peaks (*eg.* 347, 1238), and co-occurring variants are supported by the Nanopore reads. In several positions (*e.g.* 347, 407) `Gretel` can be observed to correct the reference. `Gretel` can recover enzyme isoforms from a natural microbiome.

G#	Uniprot	Protein Name Assembly Contig	Length	Forward Primer Reverse Primer ($5' \rightarrow 3'$)	PCR Outcome
G11	P45796	*Arabinoxylan arabinofuranohydrolase* Prevotella_ruminicola_23	1251	CCC TGT TCC TCT ATA CTT CGC ATG ATG C ATT TCT CAA CGC CAC CGG TCT TAC C	rRNA Contamination
G31	A7LXT3	*Xyloglucan-specific endo-beta-1,4-glucanase BoGH9A* Fibrobacter_succinogenes_subsp._succinogenes_S85	1464	GGG CAA GTA CAG CAT CAA GGT AAA CGG GTA GCT GCA ACG CTG ATC GGT GTA AG	Successful
G90	Q59219	*Intracellular exo-alpha-L-arabinofuranosidase* Prevotella_ruminicola_23	921	CAA TCG ACT TAG CTG GCA CCT TCA TGG GGT ATC TCT CTG CAC TAC TAC ACC TGC AC	Successful
G107	D5EY15	*Xylan 1,4-beta-xylosidase* Prevotella_ruminicola_23	1497	TGG TCG AAG ATT CCT TAT TCG GCT ATG TCG CTT CAG GTT GGC TGT TGC CGT CTT AC	No PCR Product
G123	P38535	*Exoglucanase XynX* Butyrivibrio_proteoclasticus_B316.combined	1425	CAC AGA CTC CTT CAT TCT TCT TCC ATG AAG ATT ATG ATA C CGC TTG CAG ATG CCT TAG CAC CAA C	Successful
G142	Q02470	*PII-type proteinase* Butyrivibrio_proteoclasticus_B316.combined	1452	ATA ACA GCT GTG CTG CCC TGA AGT G GTC ATC ATC GTT ATG GAA GGT GAT GCG	No PCR Product
G152	P15293	*PII-type proteinase* Butyrivibrio_sp._AE3009|scaffold00031	1257	TAT ATG TTG CTC AGC AGT ACG AGC CTC TTA TG CCA CGC TGA ACG AAT ACA ACC TTA CCC	Failed to amplify with Hi-Fi
G162	Q9LHE3	*Protein ASPARTIC PROTEASE IN GUARD CELL 2* lolium-draft|scaffold_3647|ref0041486	1221	CGC TCT TTC CAA CGT CGT ACA GCA C AGA AGA AGC CTA CCT TGG AGG AGC TG	Insufficient PCR Product
G165	A4J7L6	*Lon protease* Selenomonas_ruminantium_ATCC_12561|scaffold00007	1497	CAG CCA TCT GAA CCT CAA GGT GGA AAC GAA CGG CTG CGG ACA TAA GTA AGC C	No PCR Product
G251	Q73K18	*Glucose-6-phosphate isomerase* Butyrivibrio_sp._AE3009|scaffold00004	1197	CAG ATC GGT ATC GGC GGA TCA GAC C GCA AGA ACC TTA CCA AGC TGA ACA CCT TC	Successful

Table 5.7 Information on the 10 genes that were selected from 259 possible candidates for *in vitro* verification of our `Hansel` and `Gretel` haplotype recovery framework.

Chapter 6

A Rumen Haplotype Landscape

Having demonstrated in Chapter 5 that my `Hansel` and `Gretel` framework can recover and rank real haplotypes from a real microbiome; it is time to turn attention to application of the method to a data set from which we can extract new biological insight. This Chapter will provide an example of the type of biological questions we can pose, and knowledge that we can gain from metagenomic data, through the lens of the metahaplome. I will show:

- Routine mass haplotype prospecting from multiple 50 GB read sets with `Gretel`
- Generation of 663,112 haplotypes from 49,908 regions over 41 reference genomes
- Analysis of variation in the context of `EggNOG` functional categories
- Insight into the relationship between pangenomes within a microbial community
- Differences in the patterns of variation on proteins shared by different species of a community

6.1 Introduction

As part of a previous study to explore ruminant methanogenesis, conducted by Shi *et al.* [255], the rumen of 10 rams was characterised through deep whole-genome metagenomic shotgun sequencing. Two samples per ram were collected four hours after feeding (on two separate occasions separated by 14 days) before sequencing the prepared DNA with an Illumina HiSeq 2000 (2×150 bp, Section A.2.2), generating approximately 50 GB of reads per sample (1020 GB total). Whilst the original work, and derivatives [256] introduce insight to the significant sequencing effort, it is limited to descriptions of community composition. As of yet, there has been no haplotype-level investigation of the "Shi *et al.*" data set (or indeed, any other metagenomic data set). I will introduce an example of the avenues of questioning that are now open to us, when considering the variation across haplotypes of specific genes of interest, in a microbial community.

6.2 Methods

6.2.1 Existing Data

Earlier this year, Seshadri *et al.* released references for 410 cultured ruminant bacteria and archaea, a major milestone for the Hungate1000 project, whose ambitious goal is to catalogue the genomes of the rumen microbiome. It is estimated that this release covers 75% of the genus-level taxa present in the rumen [136], offering an incredibly useful starting point for future research, but the Hungate1000 cannot provide us with an insight to the true diversity of the rumen microbiome with consensus sequences alone. To overcome this for my own study, in the spirit of the human gut landscape conducted by Schloissnig *et al.* in 2013 [135] which described the diversity of the gut using data aligned to available reference genomes, I chose to conduct this "rumen landscape" using a selection of references from Hungate1000. 41 reference sequences from 8 species were selected (Table 6.1) based on those highlighted in the manuscript as having many polysaccharide utilization loci (PUL).

The Hungate collection was annotated in-house[1], using `prokka` [235], yielding a GFF file containing predicted genes for each of the reference genomes. Metagenomic reads were downloaded from the European Nucleotide Archive via Study `PRJNA202380`. Two samples (`SRR873595` & `SRR873610`) were selected from the available set of twenty (Table 6.2).

6.2.2 Alignment, Variant Calling, GFF Filtering and Haplotyping

The reads from the samples were individually aligned to each of the 41 selected genomes with `bowtie2` [217]. The intuition behind allowing a single read to map to multiple references was to overcome the potential for some regions of the union of references to act as a sink (and soak up the majority of the reads). This allows a read to contribute evidence to haplotypes on more than one reference, but arguably is a fairer strategy than allowing a read to only contribute to a single region of a single genome. Alignment produced 82 BAM files. The total number of reads and identified annotations per reference can be found in Table 6.1.

Variant calling was conducted as before with `snpper` (Section 4.1.1), generating a VCF file for every combination of sample, reference and reference contig; resulting in 5880 VCF files.

The pre-existing GFF files pertaining to the 41 chosen references were filtered (`make_bed.py`) to remove annotations shorter than 300 bp. All other regions were kept for haplotyping, and enumerated in a corresponding `BED` file. 130,311 regions were selected for haplotyping (3178$\pm$703.5 per genome). Haplotyping was carried out with `Gretel` on every combination of sample and annotation (n=260,622).

[1] By post-doctoral researcher Toby Wilkinson, as part of another project [257] who generously made it available to me

Species	Strain	Reads	Regions	Haplotypes†
Actinomyces ruminicola	*DSM*	1,399,282	553	1577
	KPR-7B	1,396,143	578	1606
Bacteroides ovatus	*NLAE-zl-C11*	2,808,062	804	3007
	NLAE-zl-C34	2,811,358	807	3173
	NLAE-zl-C500	2,020,610	772	3071
	NLAE-zl-C57	1,438,835	805	3206
Bacteroides xylanisolvens	*NLAE-zl-C182*	2,808,903	788	3171
	NLAE-zl-C202	2,004,995	788	3155
	NLAE-zl-C29	2,893,800	854	3415
	NLAE-zl-C339	2,813,310	798	3196
	NLAE-zl-G310	2,825,904	855	3200
	NLAE-zl-G339	1,875,744	847	3339
	NLAE-zl-G421	2,827,147	841	3303
Butyrivibrio proteoclasticus	*B316*	3,604,857	2393	19746
	FD2007	2,676,451	2336	19369
	P18	1,515,857	1424	10225
	P6B7	2,755,151	1160	8701
Cellulosilyticum ruminicola	*JCM_14822*	1,699,842	540	1862
Prevotella ruminicola	*AR32*	11,809,786	1233	33586
	ATCC_19189	16,532,851	2046	68864
	BPI-162	16,860,425	2084	70837
	BPI-34	17,549,238	2107	71122
	D31d	16,425,397	2088	71619
	Ga6B6	15,734,870	2045	70684
	KHT3	13,656,436	1566	45778
Pseudobutyrivibrio xylanivorans	*DSM*	1,122,435	961	6325
	DSM_14809	1,161,278	1034	6824
Ruminococcus flavefaciens	*007c*	2,387,278	1099	7184
	17	1,765,915	1054	6814
	AE3010	2,520,046	1141	7597
	ATCC_19208	456,132	550	3930
	FD-1	3,129,452	2178	15965
	MA2007	2,041,666	1056	7333
	MC2020	1,873,628	887	6105
	ND2009	2,552,008	1508	11945
	SAb67	2,814,161	1950	13886
	XPD3002	1,570,131	1144	8807
	Y1	2,107,083	1055	7333
	YAD2003	2,424,753	1072	7093
	YL228	1,785,929	1087	7827
	YRD2003	2,048,576	1020	7332

Table 6.1 The 41 Hungate references used for this rumen landscape study. Read sets from the two chosen samples were individually aligned to each of the reference sequences. For each reference, this table reports the total number of reads aligned, the number of regions that returned more than one haplotype (for dN/dS calculations) and the total number of haplotypes recovered from these regions († including duplicate haplotypes).

6.2.3 Post-processing of haplotypes

Following the execution of `Gretel` over all annotated regions, the recovered haplotypes were pooled by sample and region. That is, for each of the 130,311 candidate regions, we took the bag[2] of haplotypes recovered from the two runs of `Gretel` corresponding to the two selected samples.

As `Gretel` does not currently provide automated decision making for quality control of haplotypes[3], I conducted some conservative thresholding on the likelihood scores with the strategy depending on the number of returned haplotypes:

- **0 haplotypes**
  Regions with no haplotypes were discarded and did not contribute data to downstream analysis. A region with no returned haplotypes indicates that for both samples, it was not possible for `Gretel` to traverse the region from start to end with the evidence provided by `Hansel` via the reads, suggesting a lack of evidence in the reads, or a poor annotation.
- **1 haplotype**
  dN/dS calculations are conducted pairwise, so at least two haplotypes are required. Unfortunately, a region with only one haplotype had to be excluded from downstream analysis[4].
- **2 or 3 haplotypes**
  To avoid aggressive thresholding disadvantaging regions with few (2 or 3) haplotypes, all haplotypes were accepted for downstream analysis.
- **4 or more haplotypes**
  Given that we have previously shown that thresholding can provide a reasonable conservative basis for filtering poorer candidates; naive filtering was performed on the likelihood scores to remove all haplotypes with a likelihood worse than -1000 (the same arbitrary cutoff used to filter out the very worst HIV haplotypes in Section 4.4.2). The remaining haplotype likelihoods were scaled between 0 and 1, with the top quartile selected for downstream analysis.

Accession	Read Count	Total bp
SRR873595	$2 \times 224{,}630{,}639$	67,389,191,700
SRR873610	$2 \times 222{,}939{,}086$	66,881,725,800

Table 6.2 Metadata for samples selected to conduct the rumen landscape.

[2] Recall that a bag is a set that permits duplicates.
[3] Although, this has been identified as an area for future work (see Section 7.3)
[4] Arguably, we could have compared orphan haplotypes to the reference, but it is not a real sequence.

6.2.4 Calculation of dN/dS ratios

When describing variation observed on recovered haplotypes, there is a potential for confounding arising from differences in read coverage yielding more haplotypes for some regions across one or more genomes. To overcome this, and other bias that could arise from the underlying population size, rather than the raw numbers of recovered haplotypes, I conducted downstream analyses using the "dN/dS ratio", typically used to quantify adaptive evolution between species [258]. This ratio describes the rate of DNA mutations that cause amino acid changes in the translated sequence (non-synonymous) compared to changes that still preserve the protein sequence (synonymous). A formulation of dN/dS that is based on SNPs rather than haplotype information was recently used to characterise species diversity within the human gut [135], and the rumen [105]. Higher dN/dS ratios can imply possession of greater numbers of distinct protein isoforms for a gene [105].

For all regions with more than two haplotypes, pairwise dN/dS comparisons were performed by `CRANN` [259][5]. Ratios from all pairs were averaged, excluding cases where a dN/dS calculation was not possible (*i.e.* non-zero non-synonymous rate and synonymous rate of 0). An average dN/dS could be calculated for 49,908 regions, emitting a total of 663,112 haplotypes.

Method Summary

Existing Work

- Samples selected from the "Shi *et al.*" [255] data set; short-read Illumina sequencing from the rumen of multiple rams (50 GB per sample)
- Hungate reference genome catalogue for the rumen microbiome recently released [136]
- Hungate genomes annotated with `prokka`

Contributions

- Two samples from the "Shi *et al.*" dataset were selected to begin the rumen landscape
- 41 genomes (over 8 species) chosen from highlights of the Hungate manuscript
- `prokka` annotations filtered to be at least 300 bp, yielding 130,311 regions of interest
- Almost 900 million reads from two samples aligned against each of the 41 genomes
- `Gretel` used to recover haplotypes over the 130,311 regions, for both samples
- Haplotypes pooled and post-processed for conservative quality control
- `emapper` used to confirm haplotype reading frame and provide functional annotations
- dN/dS ratios calculated over 49,908 regions with at least 2 good-quality haplotypes

[5] Iteratively providing each region's haplotype set as a FASTA to `CRANN`'s interactive shell via Menu Option 1, and calculating pairwise distances with default parameters via Menu Option 5

6.3 Results

This work has been conducted as a pilot study, primarily to determine the computational feasibility, but also to establish the potential biological knowledge that could be gained through a full experiment to explore and exploit the haplotypes in the rumen. In the following subsections, I present several avenues of questioning that are now open to users of `Gretel`, and explore the variation observed over 663,112 haplotypes from 49,908 targeted regions from 41 different genomes.

6.3.1 Variability between species and strains

The majority of current investigations into metagenomic sequencing data attempt to describe the presence and abundance of species in a metagenome (Section 1.5). However, these approaches are typically limited to 16S rRNA gene analyses ("metataxonomics" [113]), or other marker-gene based methods (Section 2.8). With the availability of actual haplotypes, we can characterise the diversity of regions in a metagenome by calculating dN/dS ratios. Figure 6.1 describes the distribution of dN/dS ratios[6] for all regions that returned haplotypes, by species. Briefly, we can observe:

- Both a higher and broader non-synonymous variation rate for *Actinomyces ruminicola*
- Medium rate of variation between regions identified on the *Bacteroides ovatus*, *Bacteroides xylanisolvens* and *Cellulosilyticum ruminicola* genomes
- Lower and narrower variation for *Butyrivibrio proteoclasticus* and *Ruminococcus flavefaciens*
- Many outliers across all species, but generally, ratios fall between 0 and 1

However, as individual strain genomes were used to recover the haplotypes, we can go a level deeper, and compare these variation rates between and *within* species. Here, we attain **a richer form of analysis than we are currently accustomed to**: strain level analysis has only previously been possible with limited marker-gene analyses (Section 1.5). We can observe:

- Generally, variation rates across the strains of a species appears remarkably consistent
- Five strains of *Prevotella ruminicola* (dark blue) have almost exactly the same dN/dS profile; however two strains (first and last) have a larger interquartile range
- There appear to be two possible sub-groupings of *Butyrivibrio proteoclasticus* (dark green)
- Both strains of *Actinomyces ruminicola* exhibit a higher rate of variation than the other genomes
- Strains from both *Bacteroides* species appear to follow the same profile of variation

[6] To distinguish between the measured ratio of non-synonymous mutations, and the variation within the dN/dS ratios themselves, I will try to refer to "variation" and "dN/dS profiles" throughout this Chapter, respectively

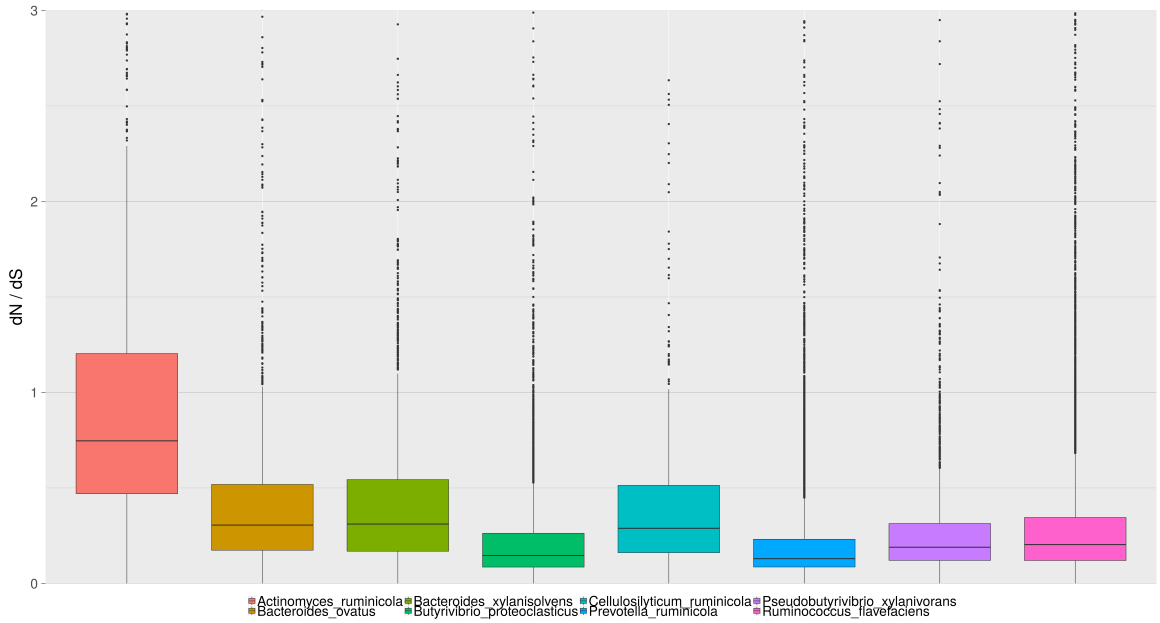


Figure 6.1 Boxplot of dN/dS ratios for the 49,908 haplotype returning regions, by species. Each box-and-whiskers describes the distribution of dN/dS ratios (y-axis) for one of eight species included in the landscape pilot, across all regions of all strains of that species (Table 6.1)

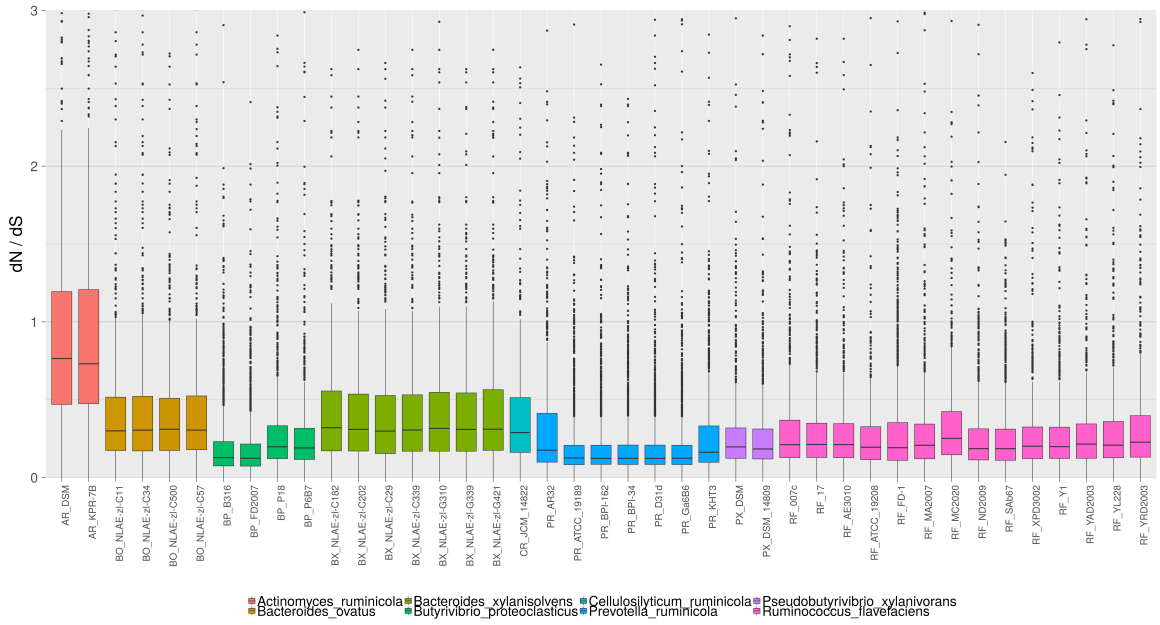


Figure 6.2 Boxplot of dN/dS ratios for 49,908 haplotype returning regions, by strain. Each box-and-whiskers describes the distribution of dN/dS ratios (y-axis) for all sets of haplotypes recovered from one of the 41 different strain genomes included in the landscape pilot (Table 6.1). Boxes are additionally colour-coded by species for readability.

6.3.2 Influence of likelihood and coverage on dN/dS

Before continuing, I wanted to briefly investigate whether there were any relationships between the observed dN/dS values, and properties of the regions from which the corresponding haplotypes were recovered. First, for every region which returned at least one haplotype, I calculated the average likelihood across all of the haplotypes, and plotted them by strain and sample in Figure 6.3.

Note that in general, haplotype likelihoods closer to 0 are "good" (*i.e.* the data observed in the `Hansel` matrix is more likely, given the haplotype is assumed to be correct), but the number of SNPs along the region, and its read depth will confound this and yield smaller likelihoods, which is why in previous comparisons of likelihood (Section 4.3.2), I scaled the data between 0 and 1. I have not performed this scaling here as I wanted to directly compare the precision (consistency) of raw likelihoods awarded by `Gretel` between the two samples used in this analysis. Thus, comparisons of the haplotype likelihoods should be directed between samples only.

Figure 6.3 appears to indicate the `Gretel`'s likelihoods are consistent between samples (and for this data set; between strains). Additionally, *Actinomyces ruminicola* haplotypes were awarded 'good' likelihoods, ruling out very bad haplotypes as the source of high rates of dN/dS in the previous plots. Similarly, I calculated the average per-base coverage of all regions that returned at least one haplotype, across both samples. The distribution of these averages are plotted in Figure 6.4. I conclude:

- Coverage appears to mimic differences observed in the dN/dS profiles between some strains of a species (Figure 6.2); namely the two apparent pairings of *B. proteoclasticus* (dark green) and the two (first and last) *P. ruminicola* (dark blue). Although generally, higher coverage yields more haplotypes, the use of dN/dS ratios should overcome bias from merely counting haplotypes[7], suggesting that coverage does not necessarily explain Figure 6.2. However, Table 6.1 reveals a difference in the number of annotated regions for these strains compared to their peers, indicating incomplete references, or an actual absence of genes between strains.
- Although it is possible for low coverage on *A. ruminicola* regions to have artificially improved likelihood scores, low coverage would also reduce the evidence available in `Hansel`, preventing recovery of spurious haplotypes that may have increased dN/dS. With literature on the species in question somewhat lacking, it is difficult to determine whether this dN/dS profile is expected.
- *P. ruminicola* is particularly well covered by the available reads, and as expected have lower likelihoods (further from 0) in comparison to other species.
- Coverage is generally consistent within a species, indicating that dN/dS profiles are unlikely to be confounded by available coverage, and do represent novel biological insight. But this uniformity could potentially be a result of all reads being allowed to align to all genomes.
- The relationship between coverage and likelihood appears consistent, indicating that future work (Section 7.3) could seek to normalise likelihoods by read depth to permit comparisons.

[7] *i.e.* dN/dS can inform us *how* the recovered sequences vary, rather than just how many there are.

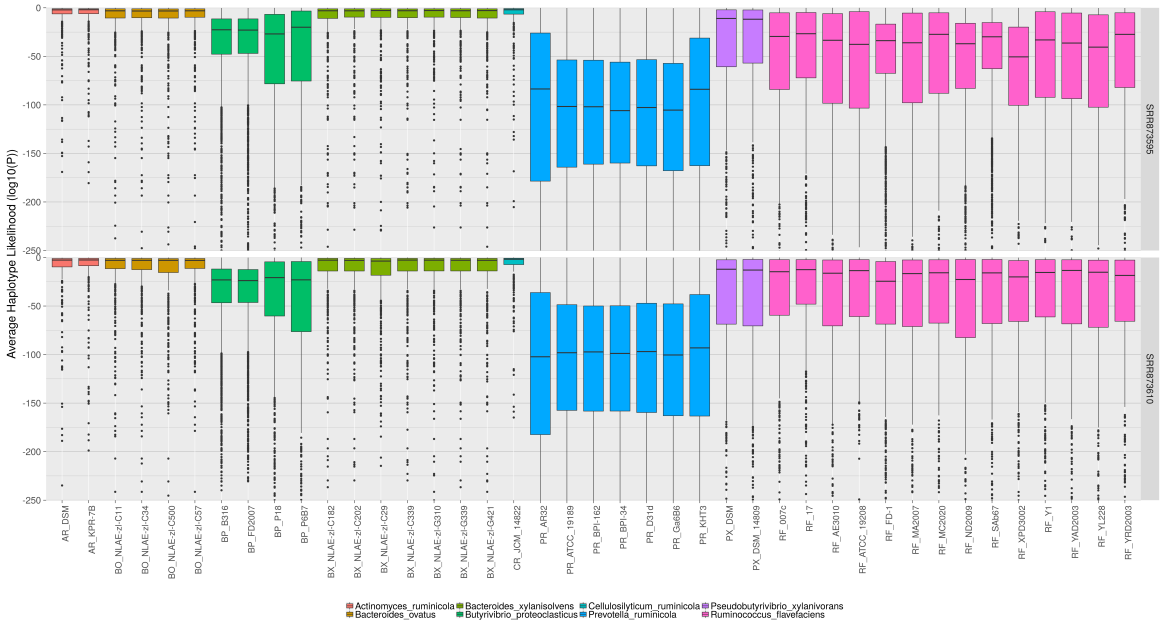


Figure 6.3 Boxplot of average haplotype likelihood for 49,908 haplotype returning regions, by strain. Each box-and-whiskers describes the distribution of average likelihood (y-axes) for all regions where haplotypes were recovered from one of the 41 different strain genomes included in the landscape pilot (Table 6.1), for each sample (row facets). Boxes are additionally colour-coded by species for readability. Haplotypes likelihoods closer to 0 are "better"; though likelihoods depend on number of SNPs, and coverage; so comparisons should be drawn between samples only.

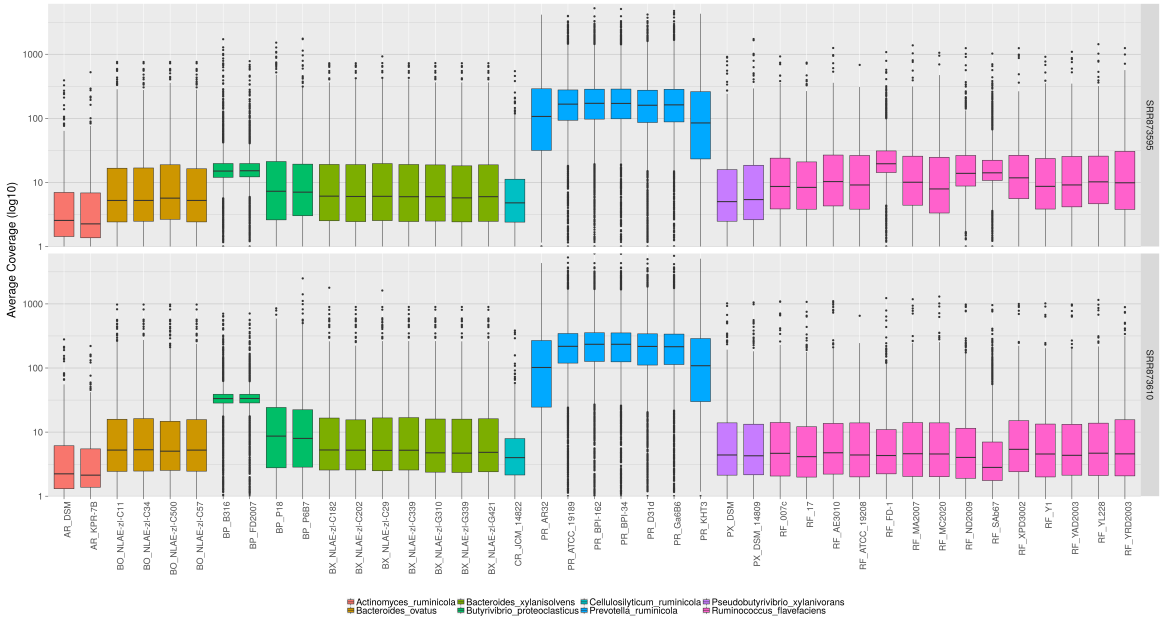


Figure 6.4 Boxplot of average read coverage for 49,908 haplotype returning regions, by strain and sample. Each box-and-whiskers describes the distribution of average read coverage (y-axes) for all regions where haplotypes were recovered from one of the 41 different strain genomes included in the landscape pilot (Table 6.1), for each sample (row facets). Boxes are additionally colour-coded by species for readability. Note that the y-axis is logarithmic.

6.3.3 Strain-level rates of variation between functional categories

Satisfied that the observed dN/dS calculations are not merely an echo of properties of the candidate regions themselves, I wanted to leverage the annotations provided by `emapper` [260] to make comparisons between the variation rates of strains, over the single-letter broad functional categories used by `EggNOG`, originally defined by Tatusov *et al.* [261, 262].

These categories can be grouped into; *Information Storage and Processing*, including translation, transcription and replication; *Cellular Processes and Signalling*, covering cell control, motility, defense, trafficking and secretions; and *Metabolism*, for the production of energy and transportation of resources and secondary metabolites. Although broad, these three top level categories can describe trends across organisms. Briefly, we can observe:

Information Storage and Processing (Figure 6.5)

- Low dN/dS amongst genes involved in translation activity, especially across *P. ruminicola* (dark blue); but with more broader profile of variation amongst the *Bacteriodetes* (orange, light green)
- Higher dN/dS on genes involved in replication (*i.e.* copying and exonuclease activities) could potentially explain the difference in profiles previously identified between the two pairs of *B. proteoclasticus* (dark green), the first and last strains of *P. ruminicola* (dark blue) in Figure 6.2, and the high dN/dS and broad profile observed on *A. ruminicola* (red)
- Very high dN/dS for both *A. ruminicola* strains, across all three information processing categories. It is unclear whether this could be attributed to the fidelity of its polymerase, or an artefact of the references themselves.

Cellular Processes and Signalling (Figure 6.6)

- Very low dN/dS amongst cell motility functions over *P. ruminicola* (dark blue), which are described as non-motile [263], and very high dN/dS for *R. flavefaciens* (pink) which are known to have between one and three flagella [264]. Oddly, *B. xylanisolvens* (orange) demonstrates high dN/dS in this category, for a species thought to be non-motile.
- Generally higher dN/dS and broader profiles amongst *B. ovatus* (orange) and *B. xylanisolvens* (green), supported by literature observing high incidences of gene transfer amongst *Bacteriodetes* in functions responsible for cell cycle control, and regulation [265]
- Uniformity in the dN/dS profiles *P. ruminicola* (dark blue)
- Higher dN/dS in chaperones amongst the *Bacteriodetes*; a category known to be impacted by changes in the diets of mice [266], and thought to be involved in cellulase regulation [267]

- Lower dN/dS and narrower profiles (with the exception of *A. ruminicola*) for critical functions such as cellular signalling, trafficking and defense

Metabolism (Function 6.7)

- Higher dN/dS for categories related to the breakdown and transfer of resources (*e.g.* energy, carbohydrates and inorganics) amongst *B. ovatus* (orange) and *B. xylanisolvens* (green), which are both known to have many glycosidehydrolases [268]. *B. xylanisolvens* is known to be involved in additional pathways capable of degrading xylans, xyloglucans and pectins [268], perhaps explaining its higher dN/dS ratios.
- Lower dN/dS in the same categories amongst *P. ruminicola* (dark blue) and *R. flavefaciens* (pink), could be related to the number of regions labelled with those functions; or potentially, a limitation in scope for amino acid changes in their proteins.
- Support for previously observed sub-groupings of *B. proteoclasticus* (pairs of dark green) and *P. ruminicola* (first and last dark blue), indicating potential niche specialisation of particular functions by strains within a species.

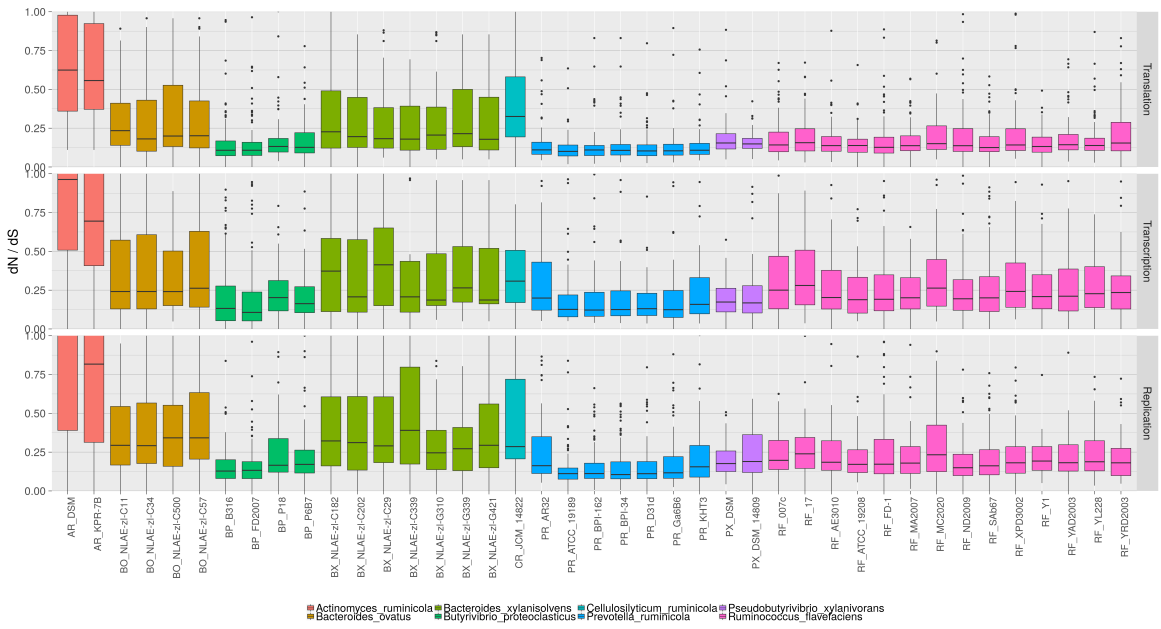


Figure 6.5 Landscape of dN/dS ratios (y-axes) across the 41 surveyed genomes for three categories of the EggNOG Information Storage and Processing classifications (row facets). From top to bottom: translation, ribosomal structure and biogenesis (J); transcription (K), and replication, recombination and repair (L). Note that the y-axes are truncated between 0 and 1.

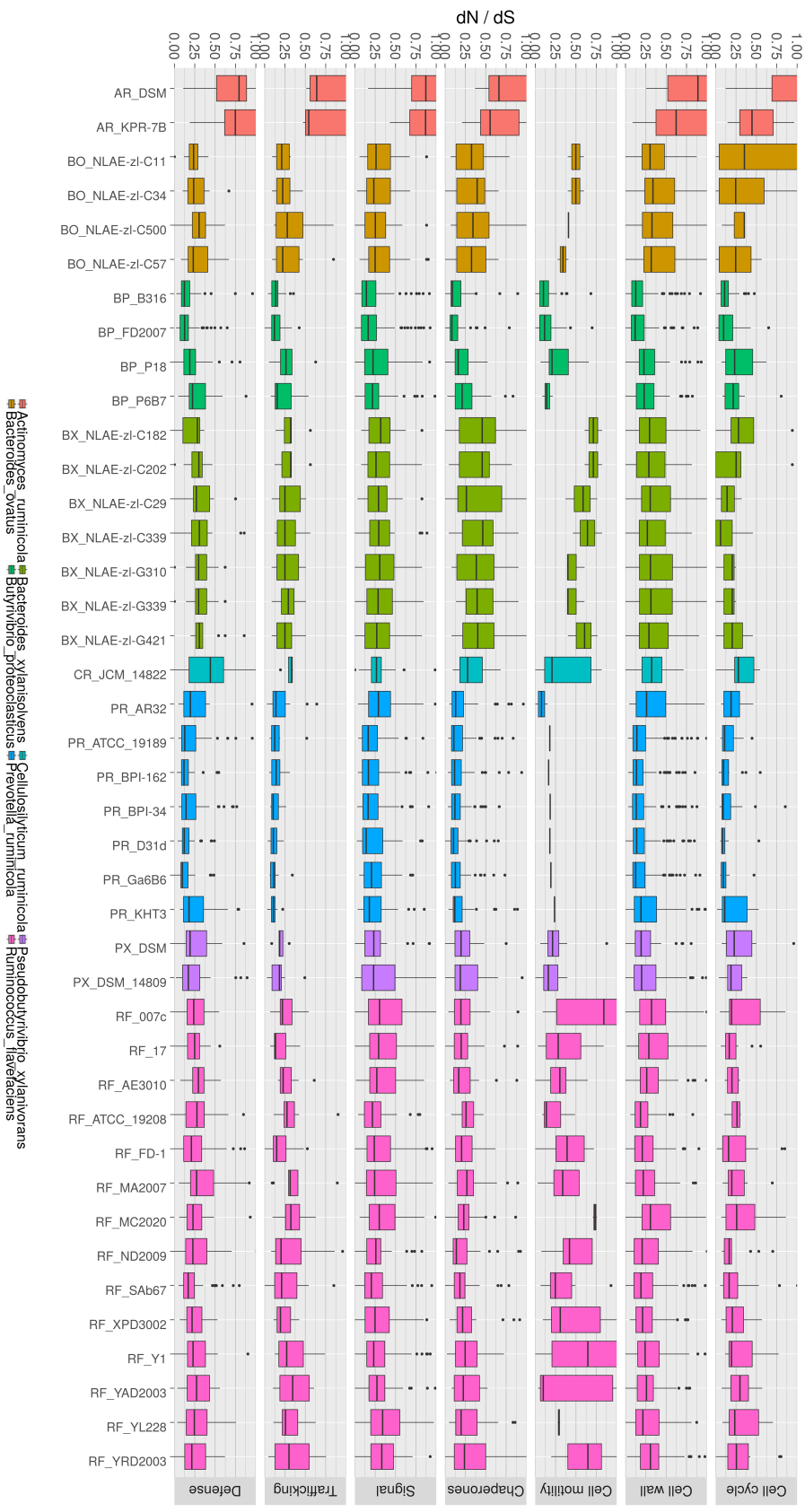


Figure 6.6 Landscape of dN/dS ratios (y-axes) across the 41 surveyed genomes for 7 categories of the EggNOG Cellular Processes and Signalling classifications (row facets). From top to bottom: cell cycle control, cell division, chromosome partitioning (D), cell wall/membrane/envelope biogenesis (M), cell motility (N), post-translational modification, protein turnover, chaperones (O), signal transduction mechanisms (T), intracellular trafficking, secretion, and vesicular transport (U), and cell defense mechanisms (V). Note that the y-axes are truncated between 0 and 1.

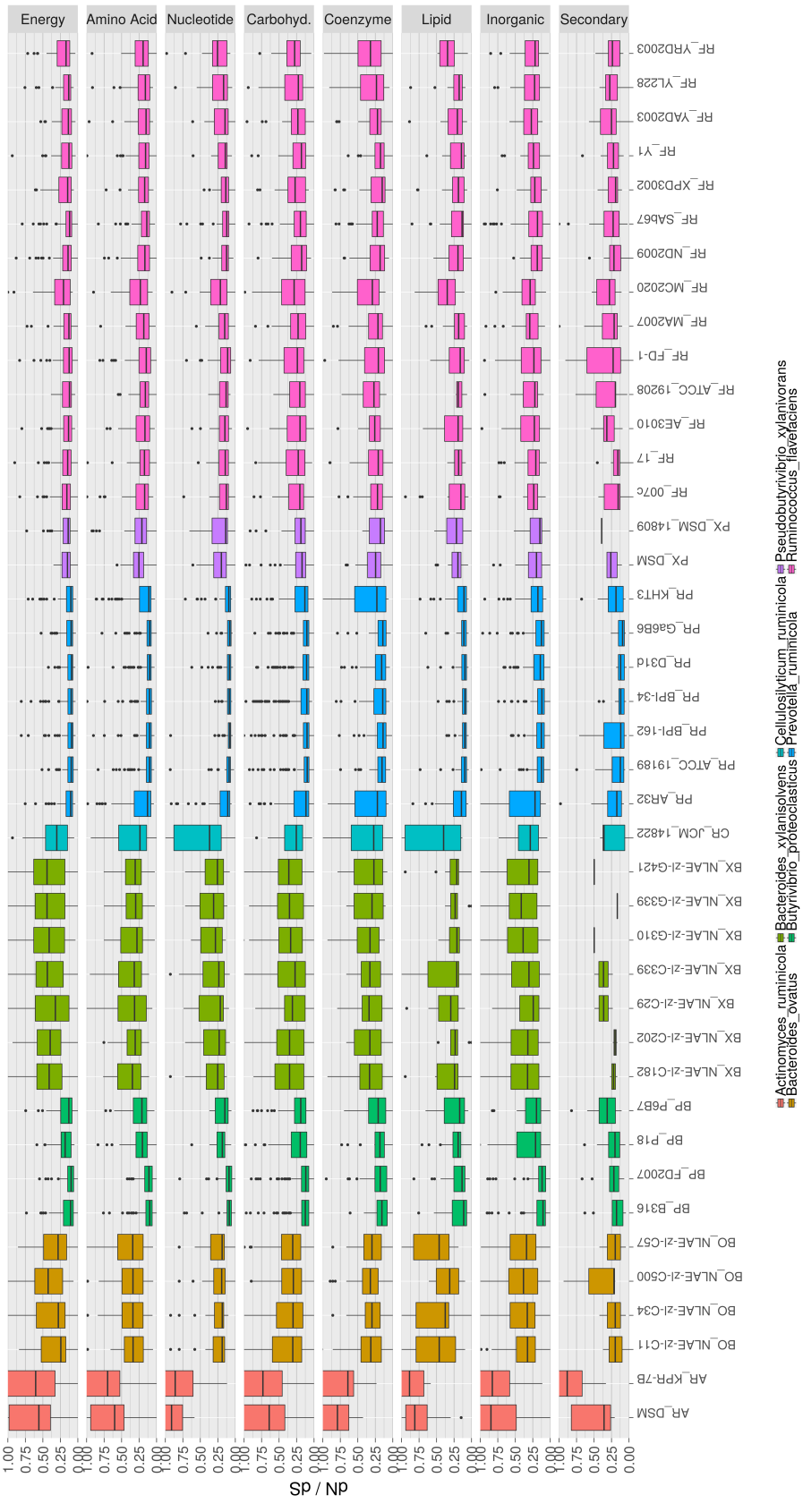


Figure 6.7 Landscape of dN/dS ratios (y-axes) across the 41 surveyed genomes for 8 categories of the EggNOG Metabolism classifications (row facets). From top to bottom: energy production and conversion (C), amino acid transport and metabolism (E), nucleotide transport and metabolism (F), carbohydrate transport and metabolism (G), coenzyme transport and metabolism (H), lipid transport and metabolism (I), inorganic transport and metabolism (P), and secondary metabolites biosynthesis, transport and catabolism (Q). Note that the y-axes are truncated between 0 and 1.

6.3.4 Characterising the variability within pangenomes

The previous sections have demonstrated variation amongst strains within a species, across broad functional categories. This population-level variation has previously been out of reach without the availability of haplotypes. Indeed, my initial plots comparing the dN/dS rates between species and strains within (Figures 6.1, 6.2) lose this subtle variation and would otherwise cause one to conclude that the strains are consistent between each other. This raises the question; *How does the observed variation fit into the pangenome?*

To address this, I identified the proportion of the 41 genomes (Table 6.1) on which each `EggNOG` non-supervised orthologous group ("NOG", *i.e.* gene families) could be found, and averaged the dN/dS for all sets of haplotypes assigned to that NOG. The proportion allows us to naively determine the "core" and "accessory" (Section 1.1) genes across the observed microbiome. Figure 6.8 illustrates how dN/dS changes at the population-level as a NOG is shared by more species and their strains. dN/dS appears to decrease as a NOG appears on more genomes, **providing preliminary evidence that there are trends within bacterial pangenomes**, and importantly; we can now observe them.

Figure 6.9 presents results for a more "traditional" species-driven view of the pangenome. We observe that *Prevotella* (pink) and *Ruminococcus* (blue), appear to follow the identified trend, but our hypothesis does not necessarily hold true for the *Bacteriodetes*, with somewhat conflicting evidence for the hypothesis demonstrated between *B. ovatus* (orange) and *B. xylanisolvens* (green), which would be an interesting starting point for a future investigation.

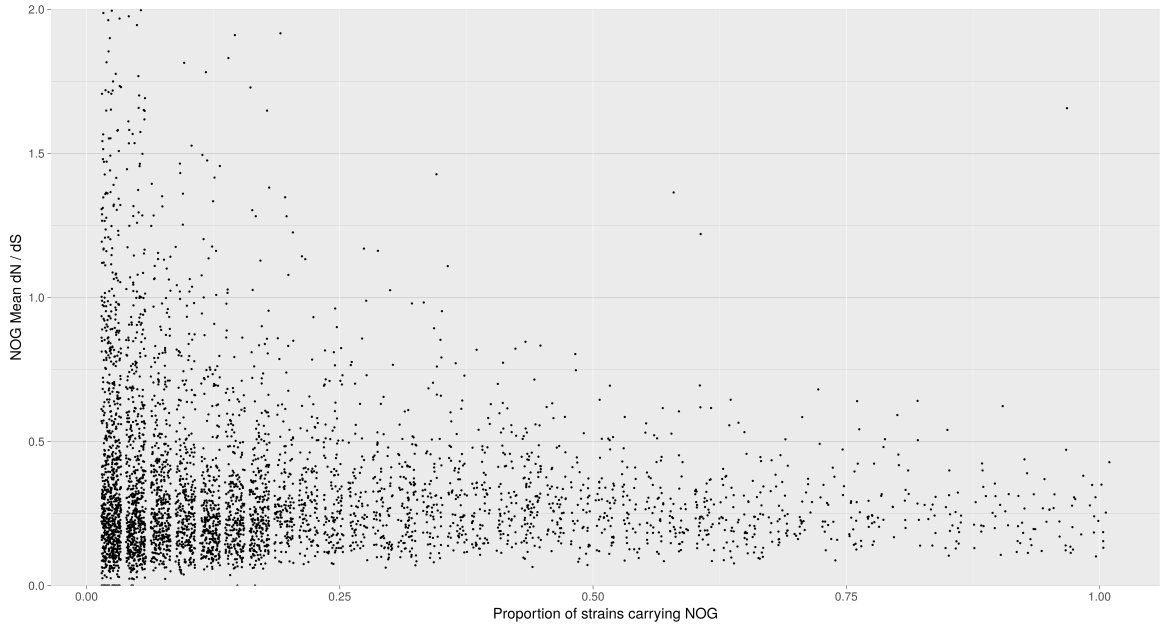


Figure 6.8 Jitter plot mapping average dN/dS variation (y-axis) over sets of haplotypes assigned to non-supervised orthologous group (NOGs), against the proportion of the 41 strains (x-axis) on which they appear.

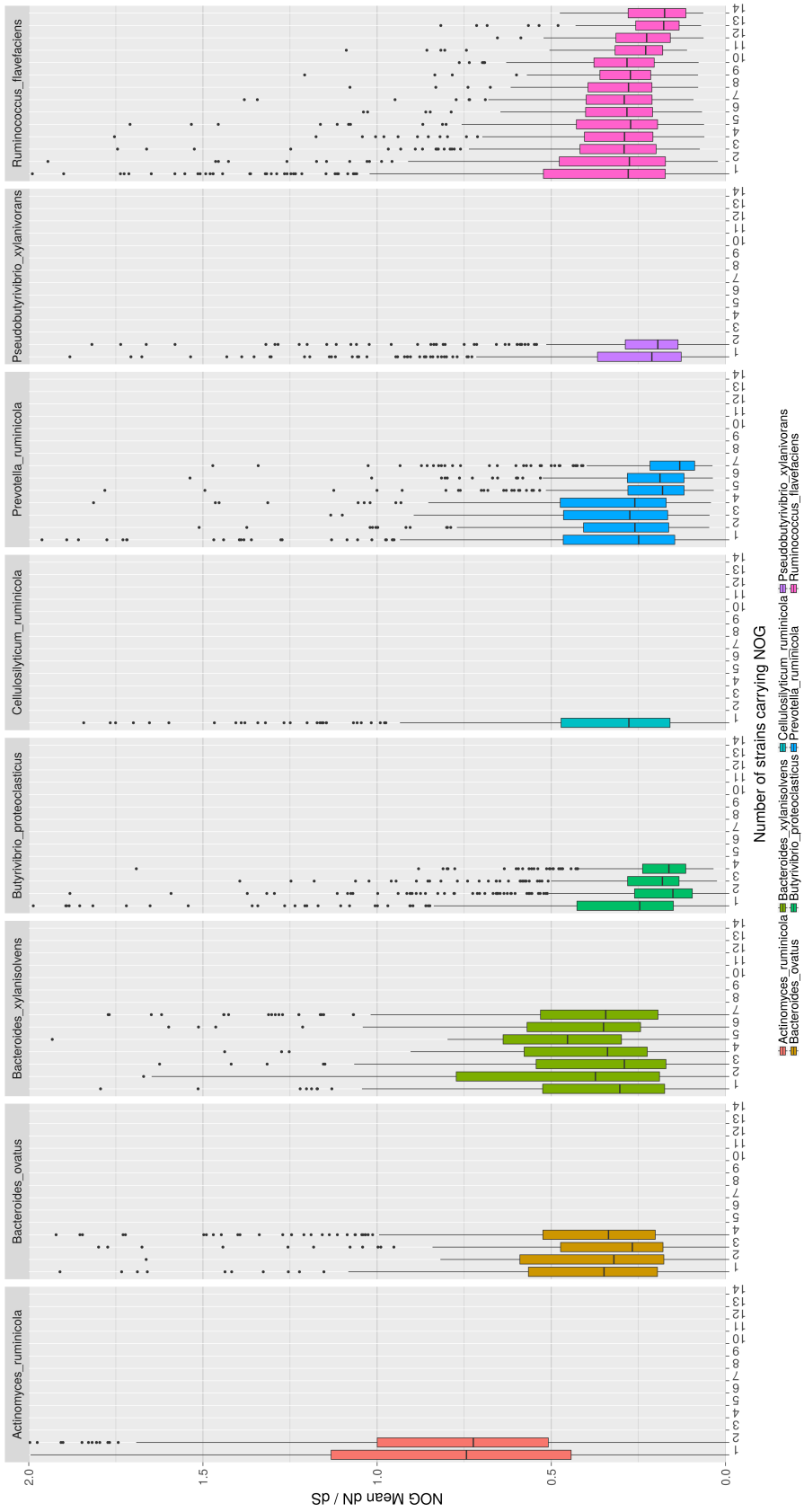


Figure 6.9 Boxplot describing the distribution of dN/dS ratios for NOGs that appear on a given number of strains for each species. Each box-and-whiskers shows the average dN/dS ratio for all haplotypes emitted from regions assigned by emapper to a particular NOG (y-axes), that appears on a given number of strains (x-axes) for each species (column facets, colour). We naively describe NOGs that appear on 1 strain (left of x-axes) as "accessory" genes, and conversely, NOGs that appear on all N strains of a species (right of x-axes) are "core".

6.3.5 Characterising the strain-level variability between gene families

With the previous section validating the existence of highly variable NOGs that are not shared by all strains of a pangenome, I now want to consider the metahaplome (Chapter 3.1) itself. In this scenario, we can envisage the metahaplome as the collection of haplotypes that code for orthologous genes across the different pangenomes (recall Figure 1.1). **My goal is to begin characterising the variation within functional elements of a microbial community, regardless of specific taxonomy**.

Here, I again leverage the availability of NOG annotations, allowing the dN/dS ratios calculated for regions across different strains and species to be considered together, under a common function. To further investigate the relationship between different pangenomes and the metahaplome, I selected the ten most frequently observed NOGs that appeared on only one (accessories), or all 14 strains (core) of *Ruminococcus flavefaciens* (Figure 6.9, pink boxplots).

R. flavefaciens Core Genome (Figure 6.11)

Figure 6.11 summarises the distribution of dN/dS ratios across the 41 different genomes, for the top ten most well represented NOGs (by raw count) that appear on all 14 of the *Ruminococcus flavefaciens* strains used in my analysis. We observe:

- Core genes for *R. flavefaciens* are not necessarily core for other species in the rumen microbiome. For example; a dockerin (NOG0XQ7Y) identified only on *R. flavefaciens*, and an integrase (COG0582) found on both *R. flavefaciens* and *B. proteoclasticus*.
- Critical functionality such as the identified ABC transporter (COG1131) can be found across all 41 genomes, generally with low dN/dS and narrow profile (with the exception of *A. ruminicola*).
- Polysaccharide utilization loci (PUL) associated with cellulases are known to include transcriptional regulation elements, including XRE regulators [269] such as NOG0XUC3. Our data potentially indicates the presence of cellulase haplotypes unique to *R. flavefaciens*.
- Some strains of *R. flavefaciens*, *P. ruminicola* and *B. proteoclasticus* demonstrate variation in flavodoxins, which have been previously shown to be involved in the cellulase degrading potential of the hindgut paunch of the termite [270].
- A wide profile of variation on a NOG associated with glycosyltransferase, especially for the two *Bacteroides* species. This is not surprising; the *Bacteroides* (and particularly *B. ovatus*) have been recognised as possessing broad glycan-degrading abilities [271].
- Variation across many species and strains for a histidine kinase, which has been hypothesised to be part of the signalling system that induces transcription of xylan utilisation genes [272]. It has particularly high dN/dS in *B. xylanisolvens*, a species known to be capable of breaking down xylan [273].

R. flavefaciens Accessory Genome (Figure 6.12)

Figure 6.12 summarises the distribution of dN/dS ratios across the 41 different genomes, for the top ten most well represented NOGs that appear on only one of the *Ruminococcus flavefaciens* strains used in my analysis, and were therefore considered to be "accessories" to its pangenome. I avoid making too many inferences from the accessory genome in this instance, as the NOGs are typically low in frequency (which explains the lack of whiskers for many of the boxes on the plot), however it is immediately recognisable that the ten selected NOGs encode for specific, less abundant functionality within the rumen microbiome. Interestingly, in some cases, **there are genes that appear on only one *R. flavefaciens* strain, but appear to be core for another species**. For example; the identified lysophospholipase (COG2755) appears on all *B. ovatus*, *B. proteoclasticus*, *B. xylanisolvens* and almost all strains of *P. ruminicola*.

I investigate this further in Figure 6.10, which compares the number of strains covered by a NOG for each species, against the number specifically for *R. flavefaciens*. Although there is indication of a core "pan-pangenome"[8]; a set of NOGs that are core to all strains of each species and *R. flavefaciens*, NOGs that are accessory to *R. flavefaciens* (towards 1 on the y-axis) appear in the core of other species (more strains on the x-axes) and vice-versa. It would appear that whether a gene is core or accessory in one species does not necessarily imply it for another.

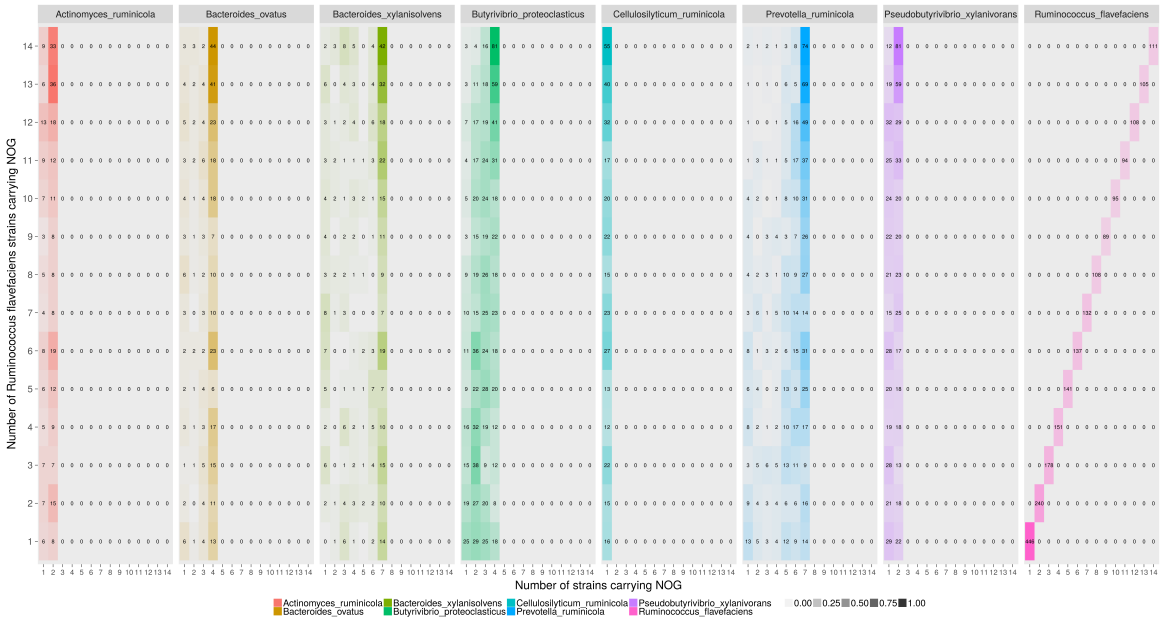


Figure 6.10 Comparison of the core and accessory genomes of the eight species. Coloured tiles represent the scaled count of NOGs that appear on a number of strains for a given species (x-axes), that also appear on a specific number of strains of *R. flavefaciens* (y-axis). Tiles are colour coded by species, and higher opacity indicates more NOGs. We see that there is evidence for a broad core pangenome (top right of each plot), but that core genes in a species do not necessarily imply they are core in another.

[8] Never quote me on that.

Figure 6.11 Landscape of dN/dS ratios (y-axes) across the 41 surveyed genomes for the 10 most abundant "core" NOGs on *R. flavefaciens*. Each box-and-whiskers describes the distribution of dN/dS ratios (y-axes) for regions annotated with one of the top 10 most abundant NOGs (row facets) by `emapper`, that appear on all 14 strains of *R. flavefaciens*. Note that the y-axes are truncated between 0 and 1.

Figure 6.12 Landscape of dN/dS ratios (y-axes) across the 41 surveyed genomes for the 10 most abundant "accessory" NOGs on *R. flavefaciens*. Each box-and-whiskers describes the distribution of dN/dS ratios (y-axes) for regions annotated with one of the top 10 most abundant NOGs (row facets) by `emapper`, that appear on only 1 strain of *R. flavefaciens*. Note that the y-axes are truncated between 0 and 1.

6.3.6 Characterising the variability within a single protein

Having demonstrated that there is variation within and between the pangenomes of species observed in a microbiome, I wanted to bring this brief exploration back to the matter at the heart of this thesis; *What variation can be observed on "exciting exploitable enzymes" across a community*?

To answer this, I was interested to follow up on some of the genes I investigated earlier as part of my *in vitro* work, in Chapter 5. I chose G90, G123 and G31 (Table 5.7) as the top three *in vitro* candidates which recruited the most nanopore reads (Table 5.3). To determine the presence of these genes in the landscape, we took the highest likelihood haplotype from each of the 49,908 regions and conducted a sequence similarity search to determine if it exhibited identity to the best haplotype recovered in Chapter 5 for G90, G123 or G31. All three genes could be found on at least one of the genomes used for this landscape pilot (Table 6.3). This is of note: **we have recovered haplotypes from the bovine rumen metatranscriptome, and used Sanger and Nanopore sequencing to demonstrate their existence (Section 5.4.2), and now we can show these haplotypes also have identity to proteins in the ovine rumen, recovered from an entirely different set of sequencing reads**.

With haplotypes, we can even begin to gain insight on what changes are happening at the protein level across a microbial community. For a collection of haplotypes corresponding to a protein predicted from the genome of one of the surveyed strains, we can attempt to calculate dN/dS ratios over small windows of the haplotypes (again, using `CRANN` [259]), to describe the distribution of alterations in the sequence that cause changes to the amino acids.

I present two such examples here. Figure 6.13 describes the variation observed over windows of the haplotypes of an endo-1,4-beta-xylanase (G123) recovered from *Prevotella ruminicola AR32*. Notably, we see amino acid changes occurring within the conserved glycoside hydrolase family 10 domain. Other variation appears to map against four known motif sites that can be used to differentiate hydrolases from this family. Additionally, a multiple sequence alignment of the recovered haplotypes (Figure 6.14) demonstrates changes in the amino acid sequence over the domain.

Figure 6.15 describes the variation across three sets of alpha-N-arabinofuranosidase (G90) haplotypes recovered from *Prevotella ruminicola ATCC-19189* (top), *Actinomyces ruminicola KPR-7B* (middle), and *Butyrivibrio proteoclasticus B316* (bottom). Interestingly, we can observe three different patterns of variation across the three species. Both *P. ruminicola* and *A. ruminicola* appear to avoid variation within the conserved Alpha-L-arabinofuranosidase, C-terminal domain. Particularly, variation of the protein in *P. ruminicola* closely flanks either side of the domain. Despite the high levels of dN/dS on the *A. ruminicola* genome throughout my previous sections, at the protein level, the variation is well delimited to the edge of the domain. *B. proteoclasticus*, a species known to possess a large suite of resource degrading enzymes of interest [274], demonstrates variation within the conserved domain.

Species	G90	G123	G31
A. ruminicola	●	○	○
B. ovatus	●	○	○
B. xylanisolvens	●	○	○
B. proteoclasticus	●	●	○
C. ruminicola	●	○	●
P. ruminicola	●	●	○
P. xylanivorans	●	○	●
R. flavefaciens	○	○	●

Table 6.3 Table indicating the presence (●) or absence (○) of Gretel candidates G90 (alpha-N-arabinofuranosidase), G123 (endo-1,4-beta-xylanase) and G31 (endocellulase) on at least one strain of each of the eight species surveyed for the landscape pilot.

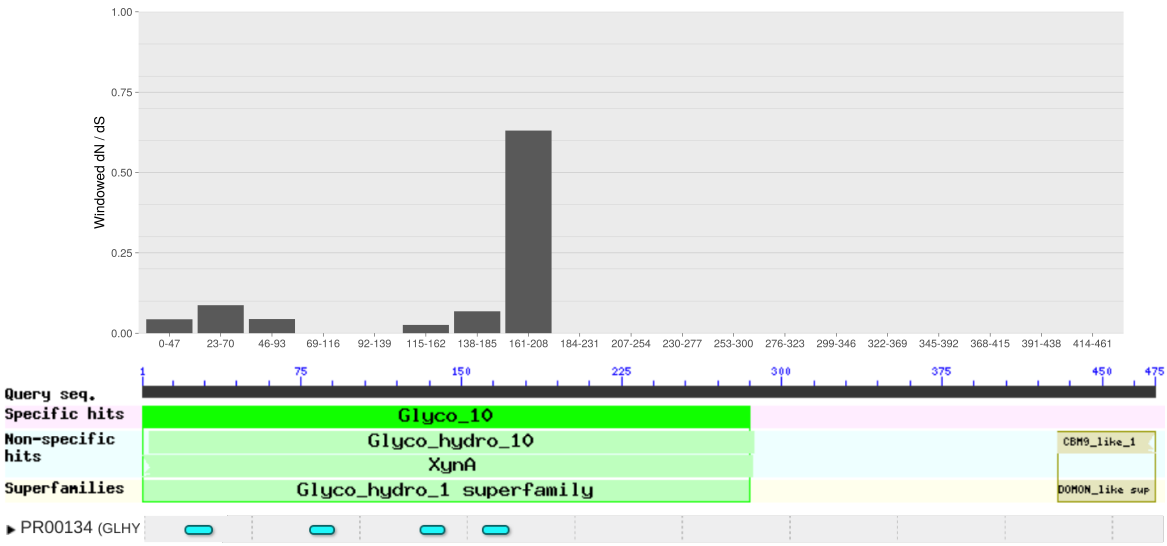


Figure 6.13 Bar plot of dN/dS windows calculated over haplotypes recovered from *Prevotella ruminicola AR32* that mapped to Gretel candidate G123 (endo-1,4-beta-xylanase; Table 5.7). Variation can be identified within the glycoside hydrolase family 10 domain's motif sites (PR00134).

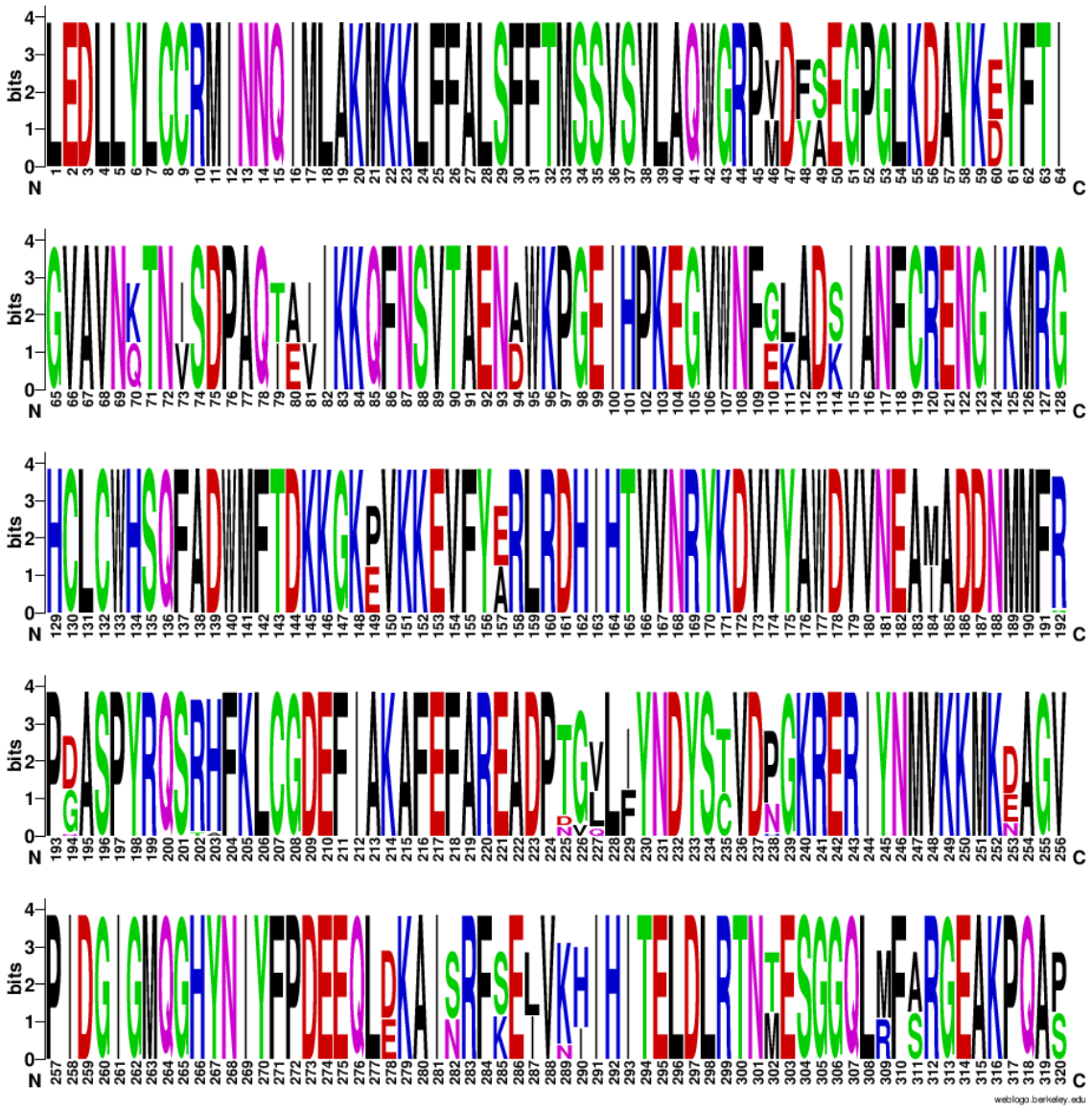


Figure 6.14 Multiple sequence alignment for G123-like sequences recovered from *Prevotella ruminicola AR32* (region 2418), expressed as a sequence logo. The plotted region is restricted to amino acids 1-320, covering the glycoside hydrolase family 10 domain. Generated via `weblogo.berkeley.edu`

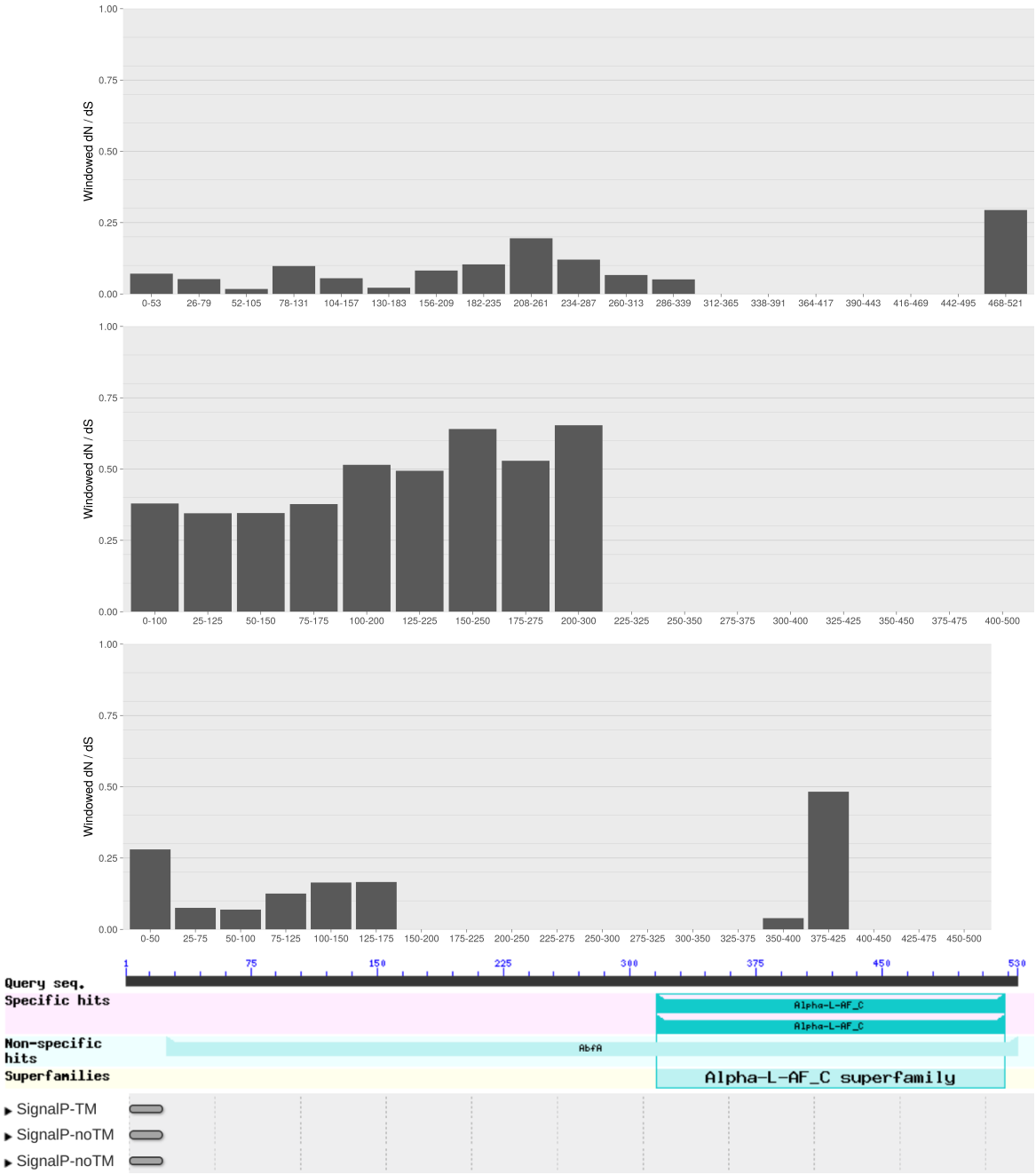


Figure 6.15 Bar plot of dN/dS windows calculated over three sets of haplotypes recovered from the ovine rumen that mapped to Gretel candidate G90 (alpha-N-arabinofuranosidase; Table 5.7). From top to bottom: *Prevotella ruminicola ATCC-19189*, *Actinomyces ruminicola KPR-7B* and *Butyrivibrio proteoclasticus B316*. We observe three different profiles of variation, with *P. ruminicola* and *A. ruminicola* avoiding amino acid changes on the Alpha-L-arabinofuranosidase, C-terminal domain, but flanking the region. Whereas *B. proteoclasticus* permits variation within this conserved site. Note that window sizes are different between examples which was required for reliable calculation of dN/dS values.

6.4 Conclusion

Throughout this whistle-stop tour of the opportunities afforded to us by the availability of haplotypes recovered from a microbial community, we have encountered many additional avenues for exploration. We have been able to describe differences in amino acid mutation rates across individual strains in a microbial community and contextualise those changes in terms of broad functionality, enzyme categories and down to specific gene families. We have been able to investigate the relationship between different pangenomes that co-exist within a microbiome, confirming that core genes in one species are not necessarily core in another, and demonstrating that we can explore the population-level variation that exists on genes across different species and their strains.

Finally, we have been able to perform windowed dN/dS calculations for specific sets of returned haplotypes, to reveal the patterns of variation observed on real proteins that are shared across strains and species, in a real natural microbiome. With the theoretical concept of the metahaplome, and a computational implementation capable of storing pairwise variation observed over aligned sequenced reads and leverage it to recover haplotypes, I argue that we now have an unprecedented level of access to the variation within a community.

It is perhaps also important to note that `Gretel` is not programmed with any biological axioms or prior knowledge, yet we can use it to recover haplotypes with amino acid variation that flanks a conserved domain, or appears in the vicinity of well-known motifs and signalling systems.

Clearly, we have many interesting starting points for further exploration of the microbiomes around us. Indeed for me, the work and results that I have introduced only briefly in this chapter were conducted as a pilot study. As of March 2018, I am working to orchestrate a full "Rumen Landscape" which aims to recover and characterise haplotypes from 1.3 million regions of interest, from 495 genomes, across 40 ovine samples. Although this is a significant undertaking, with over 26 million tasks requiring years of compute time, `Gretel` has made the recovery of metagenomic haplotypes as routine as sequence alignment. **For the first time, it is possible to characterise the diversity of groups of genes across an entire microbial population**.

Chapter 7

General Discussion and Conclusions

7.1 Performance and tractability

In Section 4.2, I demonstrated evaluation of `Hansel` and `Gretel` on haplotypes generated by simulated evolution (with `seq-gen`) in order to measure performance with regard to mutation rate, read length and coverage. Evaluation considered 6300 distinct read sets, encompassing three different read lengths, six coverage levels and seven per-haplotype mutation rates. The results (Section 4.2.2) show that `Gretel` is capable of achieving very high recovery rates, even in the presence of many SNPs. Particularly, we can observe the impact of these conditions on our ability to make effective recoveries. In general, increasing read length, read depth, and/or haplotype mutation rate can improve the quality of recovered haplotypes. We note that for some mutation rates (0.005 SNPs/hb and lower), there is insufficient evidence on the reads to recover haplotypes (recall that `Gretel` requires at least one read to span each pair of adjacent SNPs to be able to traverse the region of interest).

Section 4.3 evaluated the approach with synthetic reads generated from a metahaplome consisting of a mixture of five real *DHFR* genes, demonstrating it is possible to accurately recover real genes, even from very short reads (50 bp). This work indicated that the approach can be sensitive to the alignment of reads against the selected pseudo-reference (and the choice of that reference itself), with dropped reads affecting variant calling, and preventing pairwise SNP observations from being added to `Hansel`; which can prevent effective recovery of haplotypes with less identity to the chosen pseudo-reference (Section 4.3.3).

Having shown on smaller *in silico* data sets that `Gretel` can accurately recover haplotypes, I wanted to show that it was possible for the approach to scale appropriately to larger sequencing experiments. Section 4.4 applied `Gretel` to a challenging *de facto* viral quasispecies (Section 2.7) benchmark consisting of real Illumina sequencing reads from a laboratory mix of five distinct HIV-1 strains. Results demonstrated that `Gretel` can recover haplotypes from long (500 - 3000 bp) genes, from this complex community with 100% accuracy (Section 4.4.2).

For each of the targeted HIV genes with the exception of *env*, `Gretel` recovered at least one strain's haplotype with no error. For the highly variable envelope gene, the best reconstruction had recovered one of the strain haplotypes with the correct nucleotide for 2567 out of 2568 SNPs. Importantly, I also showed that the likelihoods that are assigned to the recovered haplotypes by `Gretel` can consistently provide a reliable ranking of haplotypes, and can be used to select the best candidates for a gene.

Continuing to scale up the evaluation strategy, Section 4.5 applies the method to a *de facto* mock microbial community. The data consists of 32 million synthetic reads, originating from 20 different genomes known to exist in the human gut. Unlike other methods that require lossy pre-processing on this data-set (Section 2.8.4), we see that without read binning or clustering, `Gretel` can recover haplotypes for hundreds of genes, shared by five *Escherichia coli* strains, from the mock community.

Most importantly, Chapter 5 provides empirical *in vitro* evidence for the performance and accuracy of `Gretel`. Having recovered haplotypes over thousands of regions of the rumen metagenome, by exploiting the evidence provided by 248 million read alignments; we chose ten particular regions of interest to investigate in the laboratory. Following gene-specific reverse transcription and PCR, I showed that predicted SNPs were supported by Sanger sequencing, and combinations of predicted SNPs (haplotypes) were supported by Nanopore sequencing. **My work presents the first biologically validated method for the recovery of haplotypes from a real microbial community**.

Finally, Chapter 6 applies `Gretel` to a large dataset, selecting 900 million reads to align to a selection of genomes known to exhibit variation that is of interest to the rumen research community [136]. For the first time, we are able to make inferences about the population-level variation within a microbiome. Initial results show that by considering the metahaplomes within the rumen, we can not only obtain novel biological insight into the variation between strains within a species, but gain understanding of the variation observed between strains and particular enzyme categories or functions, right down to the level of the genes themselves.

It is important to highlight the aforementioned discussion in Section 4.3, introducing how the approach can be sensitive to the quality of the alignments of reads against the pseudo-reference, and the choice of pseudo-reference itself. During testing I found the sensitivity was primarily an issue with read mapping; many of the synthetic reads from sequences less similar to the reference would not align back to the pseudo-reference reliably, regardless of chosen alignment software. This does raise an important caveat to this work: both assemblers and aligners will exert influence over the tractability of how many and how accurately haplotypes in a given metagenome can be recovered. Here, the discarding of reads during alignment denies `Gretel` access to critical evidence required to reconstruct those particular haplotypes.

But, it should be noted that the pseudo-reference is not used by `Hansel` or `Gretel`, it serves only as a common sequence against which to align raw reads, to provide a shared co-ordinate system for the SNPs therein. Sequences that share identity with the pseudo-reference are recovered by `Gretel` from the evidence in the `Hansel` matrix – from the evidence observed on the reads themselves. That is,

from Gretel's perspective, the reference confers no advantage to a haplotype. Very high recovery rates on sequences that share identity with the pseudo-reference are a reflection of the strength of the approach, and not a trivial recovery. A reference-free method for SNP calling, or a method that constructs or amends the reference from the SNPs [275] would be equally useful to us.

Perhaps most significantly, the tractability of the problem is bound by the quality of the data available. As originally stated by Lancia in 2001 [142], it is entirely possible that, even without error, there are scenarios where data is insufficient to successfully recover haplotypes, rendering recovery impossible.

Although my framework has been designed for the recovery of haplotypes from a region of interest in a metagenome (such as variants of a gene involved in a catalytic reaction of interest, e.g. degradation of biomass), given sufficient coverage of SNPs, our approach could work on regions significantly longer than that of a gene if desired. It could also work with data consisting of significantly longer reads. I foresee future work using complementary sequencing approaches. For example; nanopore strand sequencing (Section A.2.4) can generate a long backbone for an assembly, which can be enriched with deeper, more accurate short read sequencing (Section A.2.2) to overcome the depth and error that currently prohibits long-read technologies from becoming routine for metagenomics.

Regarding time and resource requirements, Hansel and Gretel is designed to work on all reads from a metagenome that align to some region of interest on the pseudo-reference. Typically these subsets are small (on the order of 10-100K reads) and so our framework can be run on an ordinary desktop in minutes, without significant demands on disk, memory or CPU. BAM reading is performed over multiple threads to reduce the time required to load pairwise evidence into Hansel. Run-times on data with very deep coverage, or many thousands of SNPs, such as the HIV 5-mix, are of the order of hours, but can still be executed on an ordinary desktop computer. For large scale studies, such as the one introduced by Chapter 6, as Gretel can operates on one region at a time, it is inherently parallelizable: allowing our study to be conducted over hundreds of different regions simultaneously.

7.2 Methodological comparison of our approach

In contrast to other methods (see Chapter 2), Gretel aims to make as few assumptions as possible. More importantly, our framework requires no configuration, has no parameters and is designed for metagenomic data sets where the number of haplotypes is unknown.

Most SNP calling algorithms discard SNP sites that feature three or more alleles (*i.e.* non bi-allelic sites) as errors, or under the assumption that input data represents sequenced reads from a diploid species [186]. Although ParticleHap (Section 2.5.6 [186]) acknowledges and attempts to overcome this assumption, it is only to reduce the risk of erroneously called genotypes preventing reconstruction of the two haplotypes for a diploid human genome.

A majority of the methods introduced by my earlier review attempt to optimise the popular minimum error correction (MEC, Section 2.3) criterion. These methods and their derivatives, rely on discarding or altering observed SNPs until a pair of haplotypes can be determined [144]. `Hansel` and `Gretel` use all available pairwise observations and work to recover the most likely haplotypes. Unlike other methods, we do not assume that the observed evidence must be contaminant or sequencing error that needs discarding or altering to recover the real haplotypes. Errors will be poorly supported by read data (and so will have a low probability) and are unlikely to be selected by `Gretel` to become part of a haplotype during traversal.

With little exception, haplotype reconstruction tools have been designed specifically for diploid resolution. However, the few methods that are designed for the recovery of polyploids, such as `HapCompass` (Section 2.6.1 [197]) are designed for data for single organisms, or require prior knowledge of the number of expected haplotypes [276]; or other properties of the environment, which are generally unknown for metagenomic data sets (Section 2.6).

With the advent of long-read technology, `ProbHap` recognised a niche in applying computationally expensive dynamic programming solutions to low coverage long-reads [182]. These solutions are inappropriate for high-depth short read data sets as the run time increases exponentially with coverage.

`Lens` (Section 2.6.3 [199]) is a greedy algorithm for the assembly of long-reads from overlapping short reads, with an algorithm similar to that of `FastHare` [155]. `Lens` uses a straightforward overlap assembly approach that can be used for haplotype resolution as it makes few assumptions. `Lens` generates a large number of unordered haplotypes, including many false positives, but for clean enough datasets this approach may be quick and sufficient. In our tests on the 5 HIV-1 strains benchmark, `Lens` produced 206 haplotypes for the *env* region and 147 for *pol* without any information on their ranking, whereas `Gretel` produced fewer haplotypes and provided a likelihoods as a means for selecting the best amongst them.

More recent advances in the recovery of sequences from mixed populations are limited to `ConStrains` (Section 2.8.2), `SAVAGE` (Section 2.7.3), and `DESMAN` (Section 2.8.4). `ConStrains` [216] aims to resolve strain-level differences within a set of metagenomic samples. It first uses `MetaPhlAn` to provide a species composition profile, and then chooses a corresponding set of core gene markers against which to align reads. The frequencies of SNPs in this alignment are used to cluster SNP combinations into profiles representing strains. In contrast to `Gretel`, `ConStrains` is not designed to resolve haplotypes of an enzyme or gene of interest, but instead can track strains in samples by their profiles of variation over a particular set of marker genes.

`SAVAGE` [206] is designed specifically for the related problem of viral quasi-species recovery (Section 2.7 [200]), with favourable comparison to the state-of-the-art for viral genomes. However, we note that the tool recovers many unordered haplotypes (>800 haplotypes for a lab mixture that contained just 5 strains of HIV). Additionally, as it uses an overlap assembly approach it is not particularly suited

to the complexity of metagenomes. Overlap assembly approaches such as SAVAGE and Lens [199] create large numbers of haplotypes by naively branching at choices without long range information.

Gretel overcomes the problem of recovering many possible haplotypes by outputting each haplotype with a likelihood, the probability of observing the read data given that haplotype. Haplotypes can be ordered and filtered, and likelihoods are amenable to further statistical tests. We evaluated Gretel on the same HIV data that SAVAGE reported in order to demonstrate that our method can also resolve highly variable viral quasi-species genomes. Gretel makes almost perfect recoveries from this sequenced laboratory mix of five strains (Section 4.4).

DESMAN [141] is a complex bioinformatics pipeline, that relies on read-binning, availability of good references, and a database of single-copy core species genes (SCSGs) with which to perform clustering. DESMAN then uses SNP frequencies to determine haplotypes. The use of frequencies observed across samples means that they only addressed single copy genes, as multiple copies would distort the frequencies. Furthermore, it makes use of binning software such as CONCOCT in order to filter reads before alignment to the SCSGs, and this binning process requires data from many samples (> 50 preferred). We were unable to run DESMAN on our synthetic data, which represents the scenario of analyzing genes that are not SCSGs, with diversity present in a single microbial sample. However, as demonstrated in Section 4.5, we were able to make excellent recoveries on the five *E. coli* haplotypes for 814 SCSGs (of the 982 provided) for their mock community, far more sites than DESMAN achieved.

Recovering the variation observed at the gene isoform level in a sample of a microbial community is a different problem to that of strain tracking, species binning or read clustering. Gretel provides the first practical solution to this important problem and at the same time performs as well or better than SAVAGE and DESMAN, on evaluation using their own benchmark data.

7.3 Hansel and Gretel: Future Work

Although this thesis has demonstrated Gretel's ability to recover highly accurate haplotypes from a natural microbiome, there exists room for further work. As part of future research, I intend to revisit the following aspects of the approach.

7.3.1 Reweighting

The pairwise SNP observations that contributed to the most recent haplotype are reweighted in the Hansel matrix to permit new paths to be discovered. A careful balance must be satisfied, as removing too much evidence can cause haplotypes to be "skipped", but removing too little can cause duplication, or impose a bias to generate haplotypes that are more similar to "better" haplotypes.

The current methodology seeks to remove the most recent path from the data by using the smallest ratio of evidence that contributed to that path (Section 3.3.3). This strategy appears to be a fair compromise (and my experimental work demonstrates that `Gretel` is effective), but has the potential to generate too many haplotypes if the evidence is reweighted too slowly, and conversely; too few haplotypes if the data is reweighted too quickly. This is the case if the data has very deep coverage (*e.g.* the *HIV-1* dataset, Section 4.4) or very sparse coverage (*e.g.* as recently introduced by an issue raised by a real user of my software[1]). I would like to experiment with alternative reweighting schemes. In particular, I think the read depth should contribute to the λ factor; which will address the main sensitivity.

7.3.2 Naive insertion handling

The size of the `Hansel` matrix is the square of the number of SNPs, multiplied by the square of the number of symbols. By default, there are seven symbols: the four DNA bases, the 'N', deletion '$-$' and special sentinel $\varnothing$. One could imagine that an insertion against the reference could emit a symbol of its own, a string containing the insertion itself. However, this would more than likely make the cost of storing the `Hansel` matrix in RAM too great for practical use on desktop computers. Thus due to this size constraint on the `Hansel` matrix, further thought is needed to devise a practical methodology that permits proper consideration of insertions.

An alternative approach that I would like to experiment with, is padding the reads that do not contain insertions with deletions. It turns out, that this is not a new idea, and the specification of the `SAM` (and `BAM`) format permits the use of "padded references". Although the current methodology therefore ignores insertions, it is important to note that unlike some surveyed approaches, `Gretel` does not merely discard reads containing insertions.

7.3.3 Greedy Search

As described in Section A.4, a "greedy" solution typically makes the assumption that making the best decision at each step of an algorithm will yield the best result. Indeed, we assume the "best" haplotype is the most likely haplotype, and that it can be recovered by selecting the edge with the highest probability at each SNP.

However it is possible that `Gretel` could locate solutions whose overall likelihood is comparably better if it were to make locally suboptimal choices. I would like to experiment with two ideas to investigate this. Firstly, a Metropolis-Hastings approach [165], whereby a suboptimal branch is taken over the best option with some very small probability; and secondly, implementing some form of

[1] https://github.com/SamStudio8/gretel/issues/24

"look-ahead" whereby `Gretel` calculates the probability of all paths through the next 2 or 3 nodes, and picks the route with the highest overall probability.

However, care must be taken here. The former idea would break the deterministic nature of `Gretel` (as haplotypes would depend on the stochastic decisions of the Metropolis-Hastings parameter), which is not a decision to be made lightly. The latter would increase `Gretel`'s time complexity.

7.3.4 Stopping Criterion

As briefly described in Section 3.3.4, `Gretel` will continue to generate haplotypes until either a dead end in the `Hansel` matrix is encountered, from which there is no evidence for any further transitions, or when it reaches the maximum number of haplotypes permitted by a user (which defaults to 100). In practice, I have found that this naive approach will cause `Gretel` to yield some low-quality haplotypes, especially in deep-coverage data (*e.g. HIV-1*; Section 4.4 and my landscape pilot; Chapter 6), however, `Gretel` consistently awards such haplotypes with poorer likelihoods. In both the *HIV-1* and Landscape experiments, I demonstrated the ease with which we could conservatively and effectively drop lower quality haplotypes.

As we actually have likelihoods for our haplotypes, I would like to experiment further with statistical methodologies that would allow us to test the significance of each haplotype (such as the Akaike criterion [277]), to determine whether it should be returned to the user.

7.3.5 Unused Evidence

Despite the demonstrated effectiveness of considering co-occurring SNPs on observed reads, there still remain sources of evidence not currently employed by `Gretel` – namely, paired-end reads, and per-base alignment quality scores. The vast majority of surveyed tools (Chapter 2) seek to include the evidence provided by read pairing and quality scores, which can provide key evidence that can only further improve my approach.

Indeed, taking mate pairs into account would add a wider range of non-adjacent pairwise SNP evidence to `Hansel` and is an entire dimension of knowledge we are not currently extracting. Currently, as haplotypes are built from start-to-end, one SNP at a time, it is difficult to put this evidence into use. I need to define a method that would allow us to reconcile evidence provided by a mate pair with the haplotype that must bridge the gap between the pair. I envisage a scenario whereby haplotypes are not constructed end-to-end, but rather we could determine "breakpoints" where paired end evidence is particularly strong, and reconstruct segments simultaneously with decisions in one segment triggering the use of corresponding mate observations in another.

Alternatively, I have been thinking about a form of "backpropagation", where the selection of a SNP by `Gretel` triggers the selection of one or more SNPs "further down" the haplotype SNP chain, and

we use `Gretel` to then recover SNPs that support that evidence in the reverse direction, until we reach the current forward-travelling position.

An implementation for considering quality scores is more straight-forward. We could simply weight the observation by its quality when it is inserted into `Hansel` (which does not require a pairwise observation to have an integer value). Although it is on the roadmap for a future release of `Gretel`, it was not implemented initially as quality scores for the simulated data were not generated or provided, and as `Gretel` has demonstrably worked well without it, I have not rushed for an implementation. Indeed, we do not rely on error to be identified with low quality scores, as the lack of evidence in the `Hansel` matrix is enough for `Gretel` to award it a low probability. Though, consideration of qualities may be more important for newer long-read sequencing methods that are known to generate data with higher error rates (Section A.2.3, A.2.4). The incentive to include paired-end and quality evidence is great, but it must be implemented with care.

7.3.6 Improvements to `Hansel`'s memory footprint

As described in Section 3.2, the memory requirement of `Hansel` is the product of two squares. However, for short-read data, the majority of elements in `Hansel` will be zero. `Hansel` can therefore be mathematically described as a sparse matrix, which would allow it to be significantly compressed in RAM. This is not done currently, to permit greater flexibility and ease-of-use for end users who may wish to implement `Hansel` into their own work, without `Gretel`.

However, the size of the matrix does obstruct our ability to add additional symbols; for the consideration of amino acid sequences, for example. I would like to experiment with computational methods to efficiently store sparse matrices, but still permit fast random access to its elements. An additional option may be to provide a method to calculate the size of the Markov chain's memory in advance (recall that we only consider L previously observed SNPs when predicting the next), and reduce the SNP component of the matrix from $n \times n$ to $L \times n$. This would however conflict with future plans to employ paired-end evidence.

7.3.7 Likelihood normalisation

Section 3.3.5 described the method for calculating haplotype likelihoods. As a product of the marginal SNP distributions, it can be seen that longer runs of SNPs would tend to return smaller likelihoods (multiplying more small numbers). However, generated as part of my landscape pilot, Figures 6.3 and 6.4 demonstrated an additional relationship between read coverage, and recovered haplotype likelihoods, which implies that it may be possible to scale the likelihoods by read coverage, to allow a user to compare likelihoods across different metahaplomes.

7.4 Conclusion

Earlier in my introduction, I described a core problem that currently plagues the metagenomics community: **our assemblies are not real**. Genomic assemblers are designed to generate single consensus DNA sequences representing a single individual or organism of interest [150]. Even state-of-the-art assemblers designed specifically for metagenomic data do not intend to untangle the variation between closely related species and their strains [278] but rather, setting out to generate a "backbone" (consensus) sequence that attempts to be representative of at most, a species [151].

Not only do the sequences built by an assembler represent an amalgamation of the input reads, mixing and merging overlapping sequences from different individuals and groups to create a sequence that doesn't truely exist in nature; but it actually discards the real observed variation in order to do so.

But it is this variation that we must work to characterise if we are to improve our understanding of the microbial worlds around and within us. As Huttenhower *et al.* demonstrated in 2012, the functionality of our gut microbiome is remarkably conserved, despite significant variation in community composition in hundreds of sampled individuals [145]. Our field has come to understand that it is population-level genetic variation that drives competition for niche specialisation in microbial communities [105], and this variation is widely shared within and between individuals of a microbiome through horizontal gene transfer, blurring the our long-held view of a well-defined "tree of life" [108]. Indeed, if we are to admit that the phylogenetic history of life is more of a network than a tree, it becomes difficult to imagine what meaningful insight on functional diversity could be extracted from taxonomic-centric analysis of a microbiome.

As I discussed at the opening of my thesis, there is no philosophical framework under which we can consider population-level variation of a natural microbiome. The hologenome considers the full union of genomes between a host and its attached symbiotic microbiota, the pangenome considers only the union of genomes between a species and its associated strains, and the metagenome; a somewhat overloaded term to describe the union of all genomes within a given environment or sample. Like the assemblies on which they are so dependent, these '-omes' do not allow us to investigate and uncover the variation within a single gene (or its orthologs) across an entire community.

To address this, in Section 3.1, I proposed the metahaplome: the collection of haplotypes encoding isoforms of a particular protein produced by a microbial community. If gene families are a public good [279], then the metahaplome is the marketplace on which they are traded.

Reframing the problem of uncovering variation from a microbial community as one of recovering genes and their associated isoforms – as opposed to current attempts to reconstruct entire genomes through lossy processes such as binning (Section 2.8) – has the benefit of allowing us to gain new biological insight into the true variation that drives adaption in the microbiome, whilst reducing computational complexity and our reliance on high-quality reference sequences.

Through Sections 3.2 and 3.3, I introduced `Hansel`: a data structure for storing observations of SNP pairs on sequenced reads, and `Gretel`: an algorithm that explores the evidence stored in `Hansel` as a graph to incrementally recover a sequence of SNPs that corresponds to the maximum likelihood haplotype. Together, `Hansel` and `Gretel` form a probabilistic framework to recover haplotypes from metagenomic data. I formally described the methodology we apply for the recovery of haplotypes from sequenced reads aligned to some contig in an assembly with known identity to the target gene.

Through my experimental results; I have explored the boundaries for which haplotype recovery is and is not possible (Section 4.2), showed the impact of alignment and variant calling on haplotype recovery (Section 4.3), determined that `Gretel` is robust to the noise and error of real sequencing data and has the ability to recover incredibly complex haplotypes from a viral community (Section 4.4) and scale to handle a mock microbial community, recovering haplotypes for hundreds of *E. coli* genes with high accuracy, without the need for naive pre-processing steps such as binning (Section 4.5).

In Chapter 5, I demonstrated with empirical wet laboratory evidence that `Gretel` can recover haplotypes that actually exist from a natural microbial community. **My work presents the first biologically validated method for the recovery of haplotypes from a microbial community**.

The pilot "Rumen Landscape" study presented in Chapter 6 demonstrates that the availability of haplotypes opens many different interesting avenues for research. Applying the concept of the metahaplome to a microbial community allows us to view variation through a lens of truth, revealing changes to proteins secreted by different members of the community right down at the amino acid level. We can and should now begin to look at the relationships between co-existing pangenomes within a microbial community, and seek to characterise the variation that occurs in gene families that are shared by many, or selfishly hoarded by few individuals in a microbiome.

With haplotypes harvested from large-scale sequencing projects of a community; whether the reads originate from the cow rumen, our own gut, the seas and soils, or a population of pathogenic strains, the metahaplome offers a novel, unprecedented insight to the microbial world in which we live.

The bacteria and archaea that surround us have, over billions of years, developed a war chest of enzymes to break down resources and competitors [3]. Complete characterisation of the metahaplome has significant biotechnological potential, divulging the full repertoire of enzyme isoforms for a community, which could be exploited to reduce the exponentially large search spaces and guide attempts at rational enzyme design [14], or antibiotic synthesis [32]. Comparison between recovered haplotypes can reveal the changes that adaptive evolution is in the process of fixing within a community, providing a starting point for us to rationally design and fine-tune enzymes.

Outside of industry, microbial pathogens are themselves often transmitted as a population of distinct strains [280, 281]. Understanding their variation is important for diagnosis, treatment and surveillance of infectious disease and our insight to the microbiome [282]. Our own microbiota strongly influence major aspects of our bodily development, health and wellbeing [24, 5].

Improving our understanding of the ecosystems we host would permit development of therapeutics to modify these communities to restore lost or deficient functionality, or remove members of the community causing us harm [24].

Indeed, the importance of our microbiota arises from alliances formed millions of years ago, a time throughout which we have evolved together. Microbial organisms have heavily influenced our evolutionary history [4] and still today hold the key to understanding our true origin; *Where do we fit in the network of life?* Recently, the mysterious identity of the original archaeal host cell that gave rise to the eukaryotes 1.8 billion years ago [1] became clearer: a member of the candidate superphylum of the "Asgard archaea" found in hydrothermal vent site named "Loki's Castle", between Greenland and Norway [283]. Yet our potential ancestor has only been observed using typical metagenomic analyses [284], yielding an opportunity to discover haplotypes that might shed some light on the origin of eukaryotic life.

* * *

My thesis follows in the footsteps of Lancia, who defined the problem of haplotyping the human diploid genome in 2001 [142] and Handelsman, who coined the term 'metagenomics' and argued in favour of a "culture-free" methodology that changed the way that we looked at the genomes within an environment [87]. Today, I present an argument for a similar change in thinking; that we should look to uncover the isoforms that direct and change the function of a microbial community, rather than just their taxonomy and composition. Additionally, I present a novel data structure, and an algorithm capable of leveraging the data stored within it, to recover haplotypes from sequenced reads. This thesis pushes the door to understanding microbial communities a little wider, and I hope that my arguments and evidence presented here will sway the community to adopt the concept of the metahaplome, and explore our microbial world from a population-level, haplotype-centric view.

My landscape pilot offers a brief but powerful insight into the possibilities open to us as researchers, now that we can recover haplotypes from a metagenome. My ambition is to explore these communities further, and work to catalogue the variation in gene families that exists across the population of a microbial community. The interesting, exciting isozymes that have been developed, adapted and fine-tuned by bacteria and archaea over millions of years, are no longer a trade secret.

Armed with the philosophy of the metahaplome, and the framework of Hansel and Gretel provided by this thesis, it is now possible to characterise the diversity within genes across an entire microbial population; and uncover our own evolutionary past, and the future of our industry.

Bibliography

[1] J. O. McInerney, M. J. O'connell, and D. Pisani, "The hybrid nature of the Eukaryota and a consilient view of life on Earth," *Nature Reviews Microbiology*, vol. 12, no. 6, p. 449, 2014.

[2] S. D. Dyall, M. T. Brown, and P. J. Johnson, "Ancient invasions: from endosymbionts to organelles," *Science*, vol. 304, no. 5668, pp. 253–257, 2004.

[3] N. Lane, "Bioenergetic constraints on the evolution of complex life," *Cold Spring Harbor perspectives in biology*, vol. 6, no. 5, p. a015982, 2014.

[4] M. McFall-Ngai, *et al.*, "Animals in a bacterial world, a new imperative for the life sciences," *Proceedings of the National Academy of Sciences*, vol. 110, no. 9, pp. 3229–3236, 2013.

[5] G. Rook, F. Bäckhed, B. R. Levin, M. J. McFall-Ngai, and A. R. McLean, "Evolution, human-microbe interactions, and life history plasticity," *The Lancet*, vol. 390, no. 10093, pp. 521–530, 2017.

[6] T. Domazet-Lošo and D. Tautz, "An ancient evolutionary origin of genes associated with human genetic diseases," *Molecular biology and evolution*, vol. 25, no. 12, pp. 2699–2707, 2008.

[7] R. Sorek, Y. Zhu, C. J. Creevey, M. P. Francino, P. Bork, and E. M. Rubin, "Genome-wide experimental determination of barriers to horizontal gene transfer," *Science*, vol. 318, no. 5855, pp. 1449–1452, 2007.

[8] M. J. Lercher and C. Pál, "Integration of horizontally transferred genes into regulatory interaction networks takes many million years," *Molecular biology and evolution*, vol. 25, no. 3, pp. 559–567, 2007.

[9] M. Schaechter, J. P. Williamson, J. H. Jun, and A. L. Koch, "Growth, cell and nuclear divisions in some bacteria," *Microbiology*, vol. 29, no. 3, pp. 421–434, 1962.

[10] E. Rosenberg, O. Koren, L. Reshef, R. Efrony, and I. Zilber-Rosenberg, "The role of microorganisms in coral health, disease and evolution," *Nature Reviews Microbiology*, vol. 5, no. 5, p. 355, 2007.

[11] K. R. Theis, *et al.*, "Getting the hologenome concept right: an eco-evolutionary framework for hosts and their microbiomes," *Msystems*, vol. 1, no. 2, pp. e00 028–16, 2016.

[12] R. E. Ley, C. A. Lozupone, M. Hamady, R. Knight, and J. I. Gordon, "Worlds within worlds: evolution of the vertebrate gut microbiota," *Nature Reviews Microbiology*, vol. 6, no. 10, p. 776, 2008.

[13] P. Lorenz and J. Eck, "Metagenomics and industrial applications," *Nature Reviews Microbiology*, vol. 3, no. 6, p. 510, 2005.

[14] M. Guazzaroni, A. Beloqui, J. Vieites, Y. Al-ramahi, N. Cortés, A. Ghazi, P. Golyshin, and M. Ferrer, "Metagenomic mining of enzyme diversity," in *Handbook of Hydrocarbon and Lipid Microbiology*. Springer, 2010, pp. 2911–2927.

[15] E. S. Lander, *et al.*, "Initial sequencing and analysis of the human genome." *Nature*, vol. 409, no. 6822, pp. 860–921, Feb. 2001.

[16] R. Sender, S. Fuchs, and R. Milo, "Revised estimates for the number of human and bacteria cells in the body," *PLoS biology*, vol. 14, no. 8, p. e1002533, 2016.

[17] American Academy of Microbiology, "FAQ: Human Microbiome, January 2014," Tech. Rep., 2014. [Online]. Available: http://academy.asm.org/index.php/faq-series/5122-humanmicrobiome

[18] S. R. Gill, M. Pop, R. T. Deboy, P. B. Eckburg, P. J. Turnbaugh, B. S. Samuel, J. I. Gordon, D. A. Relman, C. M. Fraser-Liggett, and K. E. Nelson, "Metagenomic analysis of the human distal gut microbiome." *Science (New York, N.Y.)*, vol. 312, no. 5778, pp. 1355–9, June 2006.

[19] P. J. Turnbaugh, R. E. Ley, M. Hamady, C. M. Fraser-Liggett, R. Knight, and J. I. Gordon, "The human microbiome project." *Nature*, vol. 449, no. 7164, pp. 804–10, Oct. 2007.

[20] F. Baquero and C. Nombela, "The microbiome as a human organ," *Clinical Microbiology and Infection*, vol. 18, no. s4, pp. 2–4, 2012.

[21] G. Clarke, R. M. Stilling, P. J. Kennedy, C. Stanton, J. F. Cryan, and T. G. Dinan, "Minireview: Gut microbiota: the neglected endocrine organ," *Molecular endocrinology*, vol. 28, no. 8, pp. 1221–1238, 2014.

[22] S. R. Bordenstein and K. R. Theis, "Host biology in light of the microbiome: ten principles of holobionts and hologenomes," *PLoS biology*, vol. 13, no. 8, p. e1002226, 2015.

[23] S. Fuentes, E. Van Nood, S. Tims, I. Heikamp-de Jong, C. J. Ter Braak, J. J. Keller, E. G. Zoetendal, and W. M. De Vos, "Reset of a critically disturbed microbial ecosystem: faecal transplant in recurrent clostridium difficile infection," *The ISME journal*, vol. 8, no. 8, p. 1621, 2014.

[24] E. Scotti, *et al.*, "Exploring the microbiome in health and disease: Implications for toxicology," *Toxicology Research and Application*, vol. 1, p. 2397847317741884, 2017.

[25] E. Patterson, P. M. Ryan, J. F. Cryan, T. G. Dinan, R. P. Ross, G. F. Fitzgerald, and C. Stanton, "Gut microbiota, obesity and diabetes," *Postgraduate medical journal*, pp. postgradmedj–2015, 2016.

[26] A. Riiser, "The human microbiome, asthma, and allergy," *Allergy, Asthma & Clinical Immunology*, vol. 11, no. 1, p. 35, 2015.

[27] C. S. Rosenfeld, "Microbiome disturbances and autism spectrum disorders," *Drug Metabolism and Disposition*, pp. dmd–115, 2015.

[28] T. R. Sampson, *et al.*, "Gut microbiota regulate motor deficits and neuroinflammation in a model of parkinson's disease," *Cell*, vol. 167, no. 6, pp. 1469–1480, 2016.

Bibliography

[29] X. C. Morgan, *et al.*, "Dysfunction of the intestinal microbiome in inflammatory bowel disease and treatment," *Genome biology*, vol. 13, no. 9, p. R79, 2012.

[30] S. Roy and G. Trinchieri, "Microbiota: a key orchestrator of cancer therapy," *Nature Reviews Cancer*, vol. 17, no. 5, p. 271, 2017.

[31] J. C. Venter, *et al.*, "Environmental genome shotgun sequencing of the Sargasso Sea," *science*, vol. 304, no. 5667, pp. 66–74, 2004.

[32] C. Zhang and S.-K. Kim, "Research and application of marine microbial enzymes: status and prospects." *Marine drugs*, vol. 8, no. 6, pp. 1920–34, Jan. 2010. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2901830&tool=pmcentrez&rendertype=abstract

[33] M. Dudek, J. Adams, M. Swain, M. Hegarty, S. Huws, and J. Gallagher, "Metaphylogenomic and Potential Functionality of the Limpet Patella pellucida's Gastrointestinal Tract Microbiome." *International journal of molecular sciences*, vol. 15, no. 10, pp. 18 819–18 839, Jan. 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25334059

[34] H. Kobayashi, Y. Hatada, T. Tsubouchi, T. Nagahama, and H. Takami, "The Hadal Amphipod Hirondellea gigas possessing a unique cellulase for digesting wooden debris buried in the deepest seafloor." *PloS one*, vol. 7, no. 8, p. e42727, Jan. 2012.

[35] J. R. Thompson, H. E. Rivera, C. J. Closek, and M. Medina, "Microbes in the coral holobiont: partners through evolution, development, and ecological interactions," *Frontiers in cellular and infection microbiology*, vol. 4, p. 176, 2015.

[36] T. P. Hughes, *et al.*, "Climate change, human impacts, and the resilience of coral reefs," *science*, vol. 301, no. 5635, pp. 929–933, 2003.

[37] I. Zilber-Rosenberg and E. Rosenberg, "Role of microorganisms in the evolution of animals and plants: the hologenome theory of evolution," *FEMS microbiology reviews*, vol. 32, no. 5, pp. 723–735, 2008.

[38] A. E. Douglas and J. H. Werren, "Holes in the hologenome: why host-microbe symbioses are not holobionts," *MBio*, vol. 7, no. 2, pp. e02 099–15, 2016.

[39] C. A. Lozupone, J. I. Stombaugh, J. I. Gordon, J. K. Jansson, and R. Knight, "Diversity, stability and resilience of the human gut microbiota," *Nature*, vol. 489, no. 7415, p. 220, 2012.

[40] M. Francino, "Antibiotics and the human gut microbiome: dysbioses and accumulation of resistances," *Frontiers in microbiology*, vol. 6, p. 1543, 2016.

[41] E. R. Mardis, "DNA sequencing technologies: 2006–2016," *Nature protocols*, vol. 12, no. 2, p. 213, 2017.

[42] J. O. McInerney, A. McNally, and M. J. O'Connell, "Why prokaryotes have pangenomes," *Nature microbiology*, vol. 2, no. 4, p. 17040, 2017.

[43] H. Tettelin, *et al.*, "Genome analysis of multiple pathogenic isolates of streptococcus agalactiae: implications for the microbial "pan-genome"," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 102, no. 39, pp. 13 950–13 955, 2005.

[44] D. Croll and B. A. McDonald, "The accessory genome as a cradle for adaptive evolution in pathogens," *PLoS Pathogens*, vol. 8, no. 4, p. e1002608, 2012.

[45] P. Lapierre and J. P. Gogarten, "Estimating the size of the bacterial pan-genome," *Trends in genetics*, vol. 25, no. 3, pp. 107–110, 2009.

[46] C. S. Smillie, M. B. Smith, J. Friedman, O. X. Cordero, L. A. David, and E. J. Alm, "Ecology drives a global network of gene exchange connecting the human microbiome," *Nature*, vol. 480, no. 7376, p. 241, 2011.

[47] R. Niehus, S. Mitri, A. G. Fletcher, and K. R. Foster, "Migration and horizontal gene transfer divide microbial genomes into multiple niches," *Nature communications*, vol. 6, p. 8924, 2015.

[48] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*, 5th ed. Garland Science, 2007, 9780815341055.

[49] F. Crick, "Central dogma of molecular biology," *Nature*, vol. 227, no. 5258, p. 561, 1970.

[50] M. E. Hibbing, C. Fuqua, M. R. Parsek, and S. B. Peterson, "Bacterial competition: surviving and thriving in the microbial jungle," *Nature Reviews Microbiology*, vol. 8, no. 1, p. 15, 2010.

[51] A. L. Demain and J. L. Adrio, "Contributions of microorganisms to industrial biology," *Molecular Biotechnology*, vol. 38, no. 1, p. 41, 2008.

[52] N. Gurung, S. Ray, S. Bose, and V. Rai, "A broader view: microbial enzymes and their relevance in industries, medicine, and beyond," *BioMed research international*, vol. 2013, 2013.

[53] R. Kohler, "The background to Eduard Buchner's discovery of cell-free fermentation," *Journal of the History of Biology*, vol. 4, no. 1, pp. 35–61, 1971.

[54] R. G. Denkewalter, D. F. Veber, F. W. Holly, and R. Hirschmann, "Total synthesis of an enzyme. I. Objective and strategy," *Journal of the American Chemical Society*, vol. 91, no. 2, pp. 502–503, 1969.

[55] B. Gutte and R. Merrifield, "The synthesis of ribonuclease A," *Journal of Biological Chemistry*, vol. 246, no. 6, pp. 1922–1941, 1971.

[56] A. Chien, D. B. Edgar, and J. M. Trela, "Deoxyribonucleic acid polymerase from the extreme thermophile Thermus aquaticus." *Journal of bacteriology*, vol. 127, no. 3, pp. 1550–1557, 1976.

[57] R. K. Saiki, D. H. Gelfand, S. Stoffel, S. J. Scharf, R. Higuchi, G. T. Horn, K. B. Mullis, and H. A. Erlich, "Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase," *Science*, vol. 239, no. 4839, pp. 487–491, 1988.

[58] A. Kumar and S. Singh, "Directed evolution: tailoring biocatalysts for industrial applications," *Critical reviews in biotechnology*, vol. 33, no. 4, pp. 365–378, 2013.

[59] L. H. van Donkelaar, J. Mostert, F. K. Zisopoulos, R. M. Boom, and A.-J. van der Goot, "The use of enzymes for beer brewing: Thermodynamic comparison on resource use," *Energy*, vol. 115, pp. 519–527, 2016.

[60] M. W. Pariza and E. A. Johnson, "Evaluating the safety of microbial enzyme preparations used in food processing: update for a new century," *Regulatory Toxicology and Pharmacology*, vol. 33, no. 2, pp. 173–186, 2001.

[61] S. R. Couto and M. A. Sanromán, "Application of solid-state fermentation to food industry—a review," *Journal of Food Engineering*, vol. 76, no. 3, pp. 291–302, 2006.

[62] M. Choct, "Enzymes for the feed industry: past, present and future," *World's Poultry Science Journal*, vol. 62, no. 1, pp. 5–16, 2006.

[63] U. C. Banerjee, R. K. Sani, W. Azmi, and R. Soni, "Thermostable alkaline protease from bacillus brevis and its characterization as a laundry detergent additive," *Process biochemistry*, vol. 35, no. 1-2, pp. 213–219, 1999.

[64] R. Araujo, M. Casal, and A. Cavaco-Paulo, "Application of enzymes for textile fibres processing," *Biocatalysis and Biotransformation*, vol. 26, no. 5, pp. 332–349, 2008.

[65] T. Tzanov, M. Calafell, G. M. Guebitz, and A. Cavaco-Paulo, "Bio-preparation of cotton fabrics," *Enzyme and Microbial Technology*, vol. 29, no. 6-7, pp. 357–362, 2001.

[66] A. Dayanandan, J. Kanagaraj, L. Sounderraj, R. Govindaraju, and G. S. Rajkumar, "Application of an alkaline protease in leather processing: an ecofriendly approach," *Journal of Cleaner Production*, vol. 11, no. 5, pp. 533–536, 2003.

[67] A. Gutiérrez, C. José, and A. T. Martínez, "Microbial and enzymatic control of pitch in the pulp and paper industry," *Applied microbiology and biotechnology*, vol. 82, no. 6, pp. 1005–1018, 2009.

[68] Y.-C. Sim, S.-G. Lee, D.-C. Lee, B.-Y. Kang, K.-M. Park, J.-Y. Lee, M.-S. Kim, I.-S. Chang, and J.-S. Rhee, "Stabilization of papain and lysozyme for application to cosmetic products," *Biotechnology letters*, vol. 22, no. 2, pp. 137–140, 2000.

[69] S. D. Minteer, B. Y. Liaw, and M. J. Cooney, "Enzyme-based biofuel cells," *Current opinion in biotechnology*, vol. 18, no. 3, pp. 228–234, 2007.

[70] S. Panke and M. Wubbolts, "Advances in biocatalytic synthesis of pharmaceutical intermediates," *Current opinion in chemical biology*, vol. 9, no. 2, pp. 188–194, 2005.

[71] S. Sanchez and A. L. Demain, "Enzymes and bioconversions of industrial, pharmaceutical, and biotechnological significance," *Organic Process Research & Development*, vol. 15, no. 1, pp. 224–230, 2010.

[72] J. L. Adrio and A. L. Demain, "Microbial enzymes: tools for biotechnological processes," *Biomolecules*, vol. 4, no. 1, pp. 117–139, 2014.

[73] A. Liese, L. Hilterhaus, G. Antranikian, and U. Kettling, *Applied Biocatalysis: From Fundamental Science to Industrial Applications*. John Wiley & Sons, 2016.

[74] M. S. Butt, M. Tahir-Nadeem, Z. Ahmad, and M. T. Sultan, "Xylanases and their applications in baking industry." *Food Technology & Biotechnology*, vol. 46, no. 1, 2008.

[75] S. Ghorai, S. P. Banik, D. Verma, S. Chowdhury, S. Mukherjee, and S. Khowala, "Fungal biotechnology in food and feed processing," *Food research international*, vol. 42, no. 5-6, pp. 577–587, 2009.

[76] A. Casaburi, R. Di Monaco, S. Cavella, F. Toldrá, D. Ercolini, and F. Villani, "Proteolytic and lipolytic starter cultures and their effect on traditional fermented sausages ripening and sensory traits," *Food Microbiology*, vol. 25, no. 2, pp. 335–347, 2008.

[77] E. Heine and H. Höcker, "Enzyme treatments for wool and cotton," *Coloration Technology*, vol. 25, no. 1, pp. 57–70, 1995.

[78] W. R. Kenealy and T. W. Jeffries, "Enzyme processes for pulp and paper: a review of recent developments." ACS Publications, 2003.

[79] A. Anwar and M. Saleemuddin, "Alkaline protease from spilosoma obliqua: potential applications in bio-formulations," *Biotechnology and applied biochemistry*, vol. 31, no. 2, pp. 85–89, 2000.

[80] A. Haddar, R. Agrebi, A. Bougatef, N. Hmidet, A. Sellami-Kamoun, and M. Nasri, "Two detergent stable alkaline serine-proteases from bacillus mojavensis a21: purification, characterization and potential application as a laundry detergent additive," *Bioresource Technology*, vol. 100, no. 13, pp. 3366–3373, 2009.

[81] *Enzyme Nomenclature 1992: Recommendations of the Nomenclature Committee of the International Union of Biochemistry and Molecular Biology and the Nomenclature and Classification of Enzymes*. Academic Press, 1992, 0122271645.

[82] R. E. Hungate, *The rumen and its microbes*. Elsevier, 2013.

[83] T. Nagaraja, "Microbiology of the rumen," in *Rumenology*. Springer, 2016, pp. 39–61.

[84] M. Hess, A. Sczyrba, R. Egan, and T. Kim, "Metagenomic discovery of biomass-degrading genes and genomes from cow rumen," *Science*, no. January, pp. 463–468, 2011. [Online]. Available: http://www.sciencemag.org/content/331/6016/463.short

[85] M. Sauer, H. Marx, and D. Mattanovich, "From rumen to industry," *Microbial cell factories*, vol. 11, no. 1, p. 121, 2012.

[86] M. Maki, K. T. Leung, and W. Qin, "The prospects of cellulase-producing bacteria for the bioconversion of lignocellulosic biomass," *International journal of biological sciences*, vol. 5, no. 5, p. 500, 2009.

[87] J. Handelsman, M. Rondon, and S. Brady, "Molecular biological access to the chemistry of unknown soil microbes: a new frontier for natural products," *Chemistry & biology*, 1998.

[88] A. C. Howe, J. K. Jansson, S. a. Malfatti, S. G. Tringe, J. M. Tiedje, and C. T. Brown, "Tackling soil diversity with the assembly of large, complex metagenomes." *Proceedings of the National Academy of Sciences of the United States of America*, vol. 111, no. 13, pp. 4904–9, Apr. 2014. [Online]. Available: http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=3977251&tool=pmcentrez&rendertype=abstract

[89] L.-X. Chen, M. Hu, L.-N. Huang, Z.-S. Hua, J.-L. Kuang, S.-J. Li, and W.-S. Shu, "Comparative metagenomic and metatranscriptomic analyses of microbial communities in acid mine drainage." *The ISME journal*, pp. 1–14, Dec. 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25535937

[90] E. Afshinnekoo, *et al.*, "Geospatial Resolution of Human and Bacterial Diversity with City-Scale Metagenomics," *Cell Systems*, pp. 1–15, Feb. 2015. [Online]. Available: http://linkinghub.elsevier.com/retrieve/pii/S2405471215000022

[91] P. D. Schloss and J. Handelsman, "Status of the microbial census," *Microbiology and Molecular Biology Reviews*, vol. 68, no. 4, pp. 686–691, 2004.

[92] P. D. Schloss, R. A. Girard, T. Martin, J. Edwards, and J. C. Thrash, "Status of the archaeal and bacterial census: an update," *MBio*, vol. 7, no. 3, pp. e00 201–16, 2016.

[93] J. Handelsman, "Metagenomics: application of genomics to uncultured microorganisms," *Microbiology and molecular biology reviews*, vol. 68, no. 4, pp. 669–685, 2004.

[94] J. T. Staley and A. Konopka, "Measurement of in situ activities of nonphotosynthetic microorganisms in aquatic and terrestrial habitats." *Annual review of microbiology*, vol. 39, pp. 321–46, Jan. 1985.

[95] D. Gevers, R. Knight, J. F. Petrosino, K. Huang, A. L. McGuire, B. W. Birren, K. E. Nelson, O. White, B. a. Methé, and C. Huttenhower, "The Human Microbiome Project: a community resource for the healthy human microbiome." *PLoS biology*, vol. 10, no. 8, p. e1001377, Jan. 2012.

[96] V. Torsvik and L. Ø vreås, "Microbial diversity and function in soil: from genes to ecosystems," *Current Opinion in Microbiology*, vol. 5, no. 3, pp. 240–245, June 2002.

[97] K. Chen and L. Pachter, "Bioinformatics for whole-genome shotgun sequencing of microbial communities," *PLoS computational biology*, vol. 1, no. 2, p. e24, 2005.

[98] K. Wetterstrand. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP). [Online]. Available: www.genome.gov/sequencingcosts [Accessed: 2017-05-10]

[99] B. Paten, A. M. Novak, J. M. Eizenga, and E. Garrison, "Genome graphs and the evolution of genome inference," *Genome research*, vol. 27, no. 5, pp. 665–676, 2017.

[100] G. Vernikos, D. Medini, D. R. Riley, and H. Tettelin, "Ten years of pan-genome analyses," *Current Opinion Microbiology*, vol. 23, pp. 148–54, 2015.

[101] R. A. Gibbs, *et al.*, "The international HapMap project," *Nature*, vol. 426, no. 6968, pp. 789–796, 2003.

[102] R. Andino and E. Domingo, "Viral quasispecies," *Virology*, vol. 479–480, pp. 46—51, 2015.

[103] F. M. Lauro, *et al.*, "The genomic basis of trophic strategy in marine bacteria," *Proceedings of the National Academy of Sciences*, vol. 106, no. 37, pp. 15 527–15 533, 2009.

[104] S. A. Huws, J. E. Edwards, C. J. Creevey, P. R. Stevens, W. Lin, S. E. Girdwood, J. A. Pachebat, and A. H. Kingston-Smith, "Temporal dynamics of the metabolically active rumen bacteria colonizing fresh perennial ryegrass," *FEMS Microbiology Ecology*, vol. 92, no. 1, p. fiv137, 2016.

[105] F. Rubino, C. Carberry, S. M. Waters, D. Kenny, M. S. McCabe, and C. J. Creevey, "Divergent functional isoforms drive niche specialisation for nutrient acquisition and use in rumen microbiome," *The ISME Journal*, 2017.

[106] J. Jung, L. Philippot, and W. Park, "Metagenomic and functional analyses of the consequences of reduction of bacterial diversity on soil functions and bioremediation in diesel-contaminated microcosms," *Nature Scientific Reports*, vol. 6, p. 23012, 2016.

[107] S. M. Nicholls, W. Aubrey, A. Edwards, K. de Grave, S. Huws, L. Schietgat, A. Soares, C. J. Creevey, and A. Clare, "Computational haplotype recovery and long-read validation identifies novel isoforms of industrially relevant enzymes from natural microbial communities," *bioRxiv*, 2018. [Online]. Available: https://www.biorxiv.org/content/early/2018/01/13/223404

[108] C. J. Creevey, D. A. Fitzpatrick, G. K. Philip, R. J. Kinsella, M. J. O'Connell, M. M. Pentony, S. A. Travers, M. Wilkinson, and J. O. McInerney, "Does a tree–like phylogeny only exist at the tips in the prokaryotes?" *Proceedings of the Royal Society of London B: Biological Sciences*, vol. 271, no. 1557, pp. 2551–2558, 2004.

[109] I. Karcagi, *et al.*, "Indispensability of horizontally transferred genes and its impact on bacterial genome streamlining," *Molecular biology and evolution*, vol. 33, no. 5, pp. 1257–1269, 2016.

[110] C. A. Hutchison, *et al.*, "Design and synthesis of a minimal bacterial genome," *Science*, vol. 351, no. 6280, p. aad6253, 2016.

[111] V. Marx, "Microbiology: the road to strain-level identification," 2016.

[112] C. Quince, A. W. Walker, J. T. Simpson, N. J. Loman, and N. Segata, "Shotgun metagenomics, from sampling to analysis," *Nature biotechnology*, vol. 35, no. 9, p. 833, 2017.

[113] J. R. Marchesi and J. Ravel, "The vocabulary of microbiome research: a proposal," 2015.

[114] R. Ranjan, A. Rani, A. Metwally, H. S. McGee, and D. L. Perkins, "Analysis of the microbiome: advantages of whole genome shotgun versus 16S amplicon sequencing," *Biochemical and biophysical research communications*, vol. 469, no. 4, pp. 967–977, 2016.

[115] C. Quast, E. Pruesse, P. Yilmaz, J. Gerken, T. Schweer, P. Yarza, J. Peplies, and F. O. Glöckner, "The SILVA ribosomal RNA gene database project: improved data processing and web-based tools," *Nucleic acids research*, vol. 41, no. D1, pp. D590–D596, 2012.

[116] S. M. Karst, M. S. Dueholm, S. J. McIlroy, R. H. Kirkegaard, P. H. Nielsen, and M. Albertsen, "Retrieval of a million high-quality, full-length microbial 16S and 18S rRNA gene sequences without primer bias," *Nature biotechnology*, 2018.

[117] S. Chakravorty, D. Helb, M. Burday, N. Connell, and D. Alland, "A detailed analysis of 16S ribosomal RNA gene segments for the diagnosis of pathogenic bacteria," *Journal of microbiological methods*, vol. 69, no. 2, pp. 330–339, 2007.

[118] S. Flygare, *et al.*, "Taxonomer: an interactive metagenomics analysis portal for universal pathogen detection and host mRNA expression profiling," *Genome biology*, vol. 17, no. 1, p. 111, 2016.

[119] D. H. Huson, S. Mitra, H.-J. Ruscheweyh, N. Weber, and S. C. Schuster, "Integrative analysis of environmental sequences using MEGAN4," *Genome research*, vol. 21, no. 9, pp. 1552–1560, 2011.

[120] D. E. Wood and S. L. Salzberg, "Kraken: ultrafast metagenomic sequence classification using exact alignments," *Genome biology*, vol. 15, no. 3, p. R46, 2014.

[121] N. Segata, L. Waldron, A. Ballarini, V. Narasimhan, O. Jousson, and C. Huttenhower, "Metagenomic microbial community profiling using unique clade-specific marker genes," *Nature methods*, vol. 9, no. 8, p. 811, 2012.

[122] J. Li, *et al.*, "An integrated catalog of reference genes in the human gut microbiome," *Nature biotechnology*, vol. 32, no. 8, p. 834, 2014.

[123] F. Rubino and C. Creevey, "MGkit: Metagenomic Framework For The Study Of Microbial Communities," 12 2014. [Online]. Available: https://figshare.com/articles/MGkit_Metagenomic_Framework_For_The_Study_Of_Microbial_Communities/1269288

[124] S. R. Eddy, "Accelerated profile HMM searches," *PLoS computational biology*, vol. 7, no. 10, p. e1002195, 2011.

[125] S. Abubucker, *et al.*, "Metabolic reconstruction for metagenomic data and its application to the human microbiome," *PLoS computational biology*, vol. 8, no. 6, p. e1002358, 2012.

[126] F. Meyer, *et al.*, "The metagenomics RAST server–a public resource for the automatic phylogenetic and functional analysis of metagenomes," *BMC bioinformatics*, vol. 9, no. 1, p. 386, 2008.

[127] R. Overbeek, T. Disz, and R. Stevens, "The SEED: a peer-to-peer environment for genome annotation," *Communications of the ACM*, vol. 47, no. 11, pp. 46–51, 2004.

[128] R. Overbeek, *et al.*, "The subsystems approach to genome annotation and its use in the project to annotate 1000 genomes," *Nucleic acids research*, vol. 33, no. 17, pp. 5691–5702, 2005.

[129] I. Pagani, K. Liolios, J. Jansson, I.-M. A. Chen, T. Smirnova, B. Nosrat, V. M. Markowitz, and N. C. Kyrpides, "The Genomes OnLine Database (GOLD) v. 4: status of genomic and metagenomic projects and their associated metadata," *Nucleic acids research*, vol. 40, no. D1, pp. D571–D579, 2011.

[130] J. Choi, *et al.*, "Strategies to improve reference databases for soil microbiomes," *The ISME journal*, vol. 11, no. 4, p. 829, 2017.

[131] A. M. Schnoes, S. D. Brown, I. Dodevski, and P. C. Babbitt, "Annotation error in public databases: misannotation of molecular function in enzyme superfamilies," *PLoS computational biology*, vol. 5, no. 12, p. e1000605, 2009.

[132] A. W. Walker, S. H. Duncan, P. Louis, and H. J. Flint, "Phylogeny, culturing, and metagenomics of the human gut microbiota," *Trends in microbiology*, vol. 22, no. 5, pp. 267–274, 2014.

[133] C. Rinke, *et al.*, "Insights into the phylogeny and coding potential of microbial dark matter," *Nature*, vol. 499, no. 7459, p. 431, 2013.

[134] Y. Marcy, *et al.*, "Dissecting biological "dark matter" with single-cell genetic analysis of rare and uncultivated TM7 microbes from the human mouth," *Proceedings of the National Academy of Sciences*, vol. 104, no. 29, pp. 11 889–11 894, 2007.

[135] S. Schloissnig, *et al.*, "Genomic variation landscape of the human gut microbiome," *Nature*, vol. 493, no. 7430, pp. 45–50, 2013.

[136] R. Seshadri, *et al.*, "Cultivation and sequencing of rumen microbiome members from the Hungate1000 Collection," *Nature biotechnology*, 2018.

[137] L. W. Hugerth, J. Larsson, J. Alneberg, M. V. Lindh, C. Legrand, J. Pinhassi, and A. F. Andersson, "Metagenome-assembled genomes uncover a global brackish microbiome," *Genome biology*, vol. 16, no. 1, p. 279, 2015.

[138] E. Garrison, J. Sirén, A. M. Novak, G. Hickey, J. M. Eizenga, E. T. Dawson, W. Jones, M. F. Lin, B. Paten, and R. Durbin, "Sequence variation aware genome references and read mapping with the variation graph toolkit," *bioRxiv*, 2017. [Online]. Available: https://www.biorxiv.org/content/early/2017/12/15/234856

[139] R. C. Elston, J. M. Satagopan, and S. Sun, "Genetic terminology," in *Statistical Human Genetics*. Springer, 2012, pp. 1–9.

[140] D. Porubsky, S. Garg, A. D. Sanders, J. O. Korbel, V. Guryev, P. M. Lansdorp, and T. Marschall, "Dense and accurate whole-chromosome haplotyping of individual genomes," *Nature communications*, vol. 8, no. 1, p. 1293, 2017.

[141] C. Quince, T. O. Delmont, S. Raguideau, J. Alneberg, A. E. Darling, G. Collins, and A. M. Eren, "Desman: a new tool for de novo extraction of strains from metagenomes," *Genome Biology*, vol. 18, no. 1, p. 181, Sep 2017. [Online]. Available: https://doi.org/10.1186/s13059-017-1309-9

[142] G. Lancia, V. Bafna, S. Istrail, R. Lippert, and R. Schwartz, "SNPs problems, complexity, and algorithms," in *Algorithms—ESA 2001*. Springer, 2001, pp. 182–193.

[143] P. Meneely, R. D. Hoang, K. Heston, and I. N. Okeke, *Genetics: Genes, Genomes, and Evolution*. Oxford University Press, 2017.

[144] G. Lancia, "Algorithmic approaches for the single individual haplotyping problem," *RAIRO-Operations Research*, vol. 50, no. 2, pp. 331–340, 2016.

[145] C. Huttenhower, *et al.*, "Structure, function and diversity of the healthy human microbiome," *Nature*, vol. 486, no. 7402, p. 207, 2012.

[146] M. Jain, *et al.*, "Nanopore sequencing and assembly of a human genome with ultra-long reads," *Nature Biotechnology*, 2018.

[147] International HapMap Consortium and others, "The international HapMap project," *Nature*, vol. 426, no. 6968, p. 789, 2003.

[148] S. McCarthy, *et al.*, "A reference panel of 64,976 haplotypes for genotype imputation," *Nature genetics*, vol. 48, no. 10, p. 1279, 2016.

[149] S. R. Browning and B. L. Browning, "Haplotype phasing: existing methods and new developments," *Nature Reviews Genetics*, vol. 12, no. 10, p. 703, 2011.

[150] J. Vollmers, S. Wiegand, and A.-K. Kaster, "Comparing and Evaluating Metagenome Assembly Tools from a Microbiologist's Perspective-Not Only Size Matters!" *PLoS One*, vol. 12, no. 1, p. e0169662, 2017.

[151] S. Nurk, D. Meleshko, A. Korobeynikov, and P. A. Pevzner, "metaSPAdes: a new versatile metagenomic assembler," *Genome research*, vol. 27, no. 5, pp. 824–834, 2017.

[152] P. Ji, Y. Zhang, J. Wang, and F. Zhao, "MetaSort untangles metagenome assembly by reducing microbial community complexity," *Nature communications*, vol. 8, p. 14306, 2017.

[153] R. Lippert, R. Schwartz, G. Lancia, and S. Istrail, "Algorithmic strategies for the single nucleotide polymorphism haplotype assembly problem," *Briefings in Bioinformatics*, vol. 3, no. 1, pp. 23–31, 2002.

[154] R. Dondi, "New results for the longest haplotype reconstruction problem," *Discrete Applied Mathematics*, vol. 160, no. 9, pp. 1299–1310, 2012.

[155] A. Panconesi and M. Sozio, "FastHare: A fast heuristic for single individual SNP haplotype reconstruction," in *International Workshop on Algorithms in Bioinformatics*. Springer, 2004, pp. 266–277.

[156] R. Cilibrasi, L. Van Iersel, S. Kelk, and J. Tromp, "On the complexity of several haplotyping problems," in *Algorithms in Bioinformatics*. Springer, 2005, pp. 128–139.

[157] F. Geraci, "A comparison of several algorithms for the single individual SNP haplotyping reconstruction problem," *Bioinformatics*, vol. 26, no. 18, pp. 2217–2225, 2010.

[158] V. Bansal and V. Bafna, "HapCUT: an efficient and accurate algorithm for the haplotype assembly problem," *Bioinformatics*, vol. 24, no. 16, pp. i153–i159, 2008.

[159] S. Levy, *et al.*, "The diploid genome sequence of an individual human," *PLoS Biol*, vol. 5, no. 10, p. e254, 2007.

[160] G. Denisov, B. Walenz, A. L. Halpern, J. Miller, N. Axelrod, S. Levy, and G. Sutton, "Consensus generation and variant detection by Celera Assembler," *Bioinformatics*, vol. 24, no. 8, pp. 1035–1040, 2008.

[161] The International HapMap Consortium, "A haplotype map of the human genome," *Nature*, vol. 437, no. 7063, pp. 1299–1320, 10 2005.

[162] L. M. Genovese, F. Geraci, and M. Pellegrini, "SpeedHap: an accurate heuristic for the single individual SNP haplotyping problem with many gaps, high reading error rate and low coverage," *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, vol. 5, no. 4, pp. 492–502, 2008.

[163] V. Bansal, A. L. Halpern, N. Axelrod, and V. Bafna, "An MCMC algorithm for haplotype assembly from whole-genome sequence data," *Genome Research*, vol. 18, no. 8, pp. 1336–1346, 2008.

[164] M. Stoer and F. Wagner, "A simple min cut algorithm," in *European Symposium on Algorithms*. Springer, 1994, pp. 141–147.

[165] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The journal of chemical physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[166] S. Lin, D. J. Cutler, M. E. Zwick, and A. Chakravarti, "Haplotype inference in random population samples," *The American Journal of Human Genetics*, vol. 71, no. 5, pp. 1129–1137, 2002.

[167] J. Duitama, T. Huebsch, G. McEwen, E.-K. Suk, and M. R. Hoehe, "ReFHap: a reliable and fast algorithm for single individual haplotyping," in *Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*. ACM, 2010, pp. 160–169.

[168] D. Aguiar and S. Istrail, "HapCompass: a fast cycle basis algorithm for accurate haplotype assembly of sequence data," *Journal of Computational Biology*, vol. 19, no. 6, pp. 577–590, 2012.

[169] A. McKenna, *et al.*, "The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data," *Genome research*, vol. 20, no. 9, pp. 1297–1303, 2010.

[170] M. Xie, J. Wang, and T. Jiang, "A fast and accurate algorithm for single individual haplotyping," *BMC Systems Biology*, vol. 6, no. Suppl 2, p. S8, 2012.

[171] L. M. Li, J. H. Kim, and M. S. Waterman, "Haplotype reconstruction from SNP alignment," *Journal of Computational Biology*, vol. 11, no. 2-3, pp. 505–516, 2004.

[172] G. A. Churchill and M. S. Waterman, "The accuracy of DNA sequences: estimating sequence quality," *Genomics*, vol. 14, no. 1, pp. 89–98, 1992.

[173] G. A. Churchill, "Stochastic models for heterogeneous DNA sequences," *Bulletin of mathematical biology*, vol. 51, no. 1, pp. 79–94, 1989.

[174] R.-S. Wang, L.-Y. Wu, X.-S. Zhang, and L. Chen, "A markov chain model for haplotype assembly from SNP fragments," *Genome Informatics*, vol. 17, no. 2, pp. 162–171, 2006.

[175] L. Eronen, F. Geerts, and H. Toivonen, "A markov chain approach to reconstruction of long haplotypes," in *Biocomputing 2004*. World Scientific, 2003, pp. 104–115.

[176] P. Rastas, M. Koivisto, H. Mannila, and E. Ukkonen, "A hidden markov technique for haplotype reconstruction," in *International Workshop on Algorithms in Bioinformatics*. Springer, 2005, pp. 140–151.

[177] J. H. Kim, M. S. Waterman, and L. M. Li, "Diploid genome reconstruction of Ciona intestinalis and comparative analysis with Ciona savignyi," *Genome research*, vol. 17, no. 7, pp. 1101–1110, 2007.

[178] B. Ewing, L. Hillier, M. C. Wendl, and P. Green, "Base-calling of automated sequencer traces using Phred. I. Accuracy assessment," *Genome research*, vol. 8, no. 3, pp. 175–185, 1998.

[179] H. Matsumoto and H. Kiryu, "MixSIH: a mixture model for single individual haplotyping," *BMC genomics*, vol. 14, no. Suppl 2, p. S5, 2013.

[180] C. Shalizi, "36-754, Advanced Probability II or Almost None of Stochastic Processes," http://www.stat.cmu.edu/~cshalizi/754/2006/notes/all.pdf, 2006.

[181] D. J. MacKay, *Information theory, inference and learning algorithms*. Cambridge University Press, 2003.

[182] V. Kuleshov, "Probabilistic single-individual haplotyping," *Bioinformatics*, vol. 30(17), pp. i379–85, 2014.

[183] A. McCallum, "Graphical Models. Lecture 11: Clique Trees," https://people.cs.umass.edu/~mccallum/courses/gm2011/11-clique-trees.pdf, 2011.

[184] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques - Adaptive Computation and Machine Learning*. MIT Press, 2009.

[185] T. W. Fawcett and A. D. Higginson, "Heavy use of equations impedes communication among biologists," *Proceedings of the National Academy of Sciences*, vol. 109, no. 29, pp. 11 735–11 739, 2012.

[186] S. Ahn and H. Vikalo, "Joint haplotype assembly and genotype calling via sequential Monte Carlo algorithm," *BMC Bioinformatics*, vol. 16, no. 1, p. 223, 2015.

[187] 1000 Genomes Project Consortium and others, "An integrated map of genetic variation from 1,092 human genomes," *Nature*, vol. 491, no. 7422, p. 56, 2012.

[188] P. Edge, V. Bafna, and V. Bansal, "HapCUT2: robust and accurate haplotype assembly for diverse sequencing technologies," *Genome research*, vol. 27, no. 5, pp. 801–812, 2017.

[189] S. Beretta, M. Patterson, T. Marschall, M. Martin, S. Zaccaria, G. Della Vedova, and P. Bonizzoni, "HapCHAT: Adaptive haplotype assembly for efficiently leveraging high coverage in long reads," *bioRxiv*, 2018. [Online]. Available: https://www.biorxiv.org/content/early/2018/02/12/170225

[190] M. Patterson, T. Marschall, N. Pisanti, L. Van Iersel, L. Stougie, G. W. Klau, and A. Schönhuth, "WhatsHap: weighted haplotype assembly for future-generation sequencing reads," *Journal of Computational Biology*, vol. 22, no. 6, pp. 498–509, 2015.

[191] Y. Pirola, S. Zaccaria, R. Dondi, G. W. Klau, N. Pisanti, and P. Bonizzoni, "HapCol: accurate and memory-efficient haplotype assembly from long reads," *Bioinformatics*, vol. 32, no. 11, pp. 1610–1617, 2015.

[192] H. J. Greenberg, W. E. Hart, and G. Lancia, "Opportunities for combinatorial optimization in computational biology," *INFORMS Journal on Computing*, vol. 16, no. 3, pp. 211–231, 2004.

[193] R. J. Roberts, M. O. Carneiro, and M. C. Schatz, "The advantages of SMRT sequencing," *Genome biology*, vol. 14, no. 6, p. 405, 2013.

[194] C. R. O'Donnell, H. Wang, and W. B. Dunbar, "Error analysis of idealized nanopore sequencing," *Electrophoresis*, vol. 34, no. 15, pp. 2137–2144, 2013.

[195] M. O. Carneiro, C. Russ, M. G. Ross, S. B. Gabriel, C. Nusbaum, and M. A. DePristo, "Pacific biosciences sequencing technology for genotyping and variation discovery in human data," *BMC genomics*, vol. 13, no. 1, p. 375, 2012.

[196] M. C. Stancu, *et al.*, "Mapping and phasing of structural variation in patient genomes using nanopore sequencing," *Nature Communications*, vol. 8, no. 1, p. 1326, 2017.

[197] D. Aguiar and S. Istrail, "Haplotype assembly in polyploid genomes and identical by descent shared tracts," *Bioinformatics*, vol. 29, no. 13, pp. i352–i360, 2013.

[198] E. Berger, D. Yorukoglu, J. Peng, and B. Berger, "Haptree: A novel bayesian framework for single individual polyplotyping using ngs data," *PLoS computational biology*, vol. 10, no. 3, p. e1003502, 2014.

[199] V. Kuleshov, C. Jiang, W. Zhou, F. Jahanbani, S. Batzoglou, and M. Snyder, "Synthetic long-read sequencing reveals intraspecies diversity in the human microbiome," *Nature Biotechnology*, vol. 34, p. 64 – 69, 2016.

[200] R. Rose, B. Constantinides, A. Tapinos, D. L. Robertson, and M. Prosperi, "Challenges in the analysis of viral metagenomes," *Virus Evolution*, vol. 2, no. 2, p. vew022, 2016.

[201] O. Zagordi, A. Bhattacharya, N. Eriksson, and N. Beerenwinkel, "ShoRAH: estimating the genetic diversity of a mixed sample from next-generation sequencing data," *BMC bioinformatics*, vol. 12, no. 1, p. 119, 2011.

[202] O. Zagordi, L. Geyrhofer, V. Roth, and N. Beerenwinkel, "Deep sequencing of a genetically heterogeneous sample: local haplotype reconstruction and read error correction," *Journal of computational biology*, vol. 17, no. 3, pp. 417–428, 2010.

[203] M. Schirmer, W. T. Sloan, and C. Quince, "Benchmarking of viral haplotype reconstruction programmes: an overview of the capacities and limitations of currently available programmes," *Briefings in bioinformatics*, vol. 15, no. 3, pp. 431–442, 2012.

[204] M. C. Prosperi, L. Yin, D. J. Nolan, A. D. Lowe, M. M. Goodenow, and M. Salemi, "Empirical validation of viral quasispecies assembly algorithms: state-of-the-art and challenges," *Scientific Reports*, vol. 3, p. 2837, 2013.

[205] S. Pulido-Tamayo, A. Sánchez-Rodríguez, T. Toon Swings, B. Van den Bergh, A. Dubey, H. Steenackers, J. Michiels, J. Fostier, and K. Marchal, "Frequency-based haplotype reconstruction from deep sequencing data of bacterial populations," *Nucleic Acids Res*, vol. 43(16), p. e105, 2015.

[206] J. A. Baaijens, A. Z. El Aabidine, E. Rivals, and A. Schönhuth, "De novo assembly of viral quasispecies using overlap graphs," *Genome Research*, vol. 27, no. 5, pp. 835–848, 2017.

[207] R. Malhotra, M. M. S. Wu, A. Rodrigo, M. Poss, and R. Acharya, "Maximum likelihood de novo reconstruction of viral populations using paired end sequencing data," *arXiv preprint arXiv:1502.04239*, 2015.

[208] A. Töpfer, T. Marschall, R. A. Bull, F. Luciani, A. Schönhuth, and N. Beerenwinkel, "Viral quasispecies assembly via maximal clique enumeration," *PLoS computational biology*, vol. 10, no. 3, p. e1003515, 2014.

[209] J. Chen, Y. Zhao, and Y. Sun, "De novo haplotype reconstruction in viral quasispecies using paired-end read guided path finding," *bioRxiv*, 2018. [Online]. Available: https://www.biorxiv.org/content/early/2018/01/28/254987

[210] J. Baaijens, B. van der Roest, J. Koester, L. Stougie, and A. Schoenhuth, "Full-length de novo viral quasispecies assembly through variation graph construction," *bioRxiv*, 2018. [Online]. Available: https://www.biorxiv.org/content/early/2018/03/23/287177

[211] S. Barik, S. Das, and H. Vikalo, "QSdpR: Viral quasispecies reconstruction via correlation clustering," *Genomics*, 2017.

[212] A. Kislyuk, S. Bhatnagar, J. Dushoff, and J. S. Weitz, "Unsupervised statistical clustering of environmental shotgun sequences," *BMC bioinformatics*, vol. 10, no. 1, p. 316, 2009.

[213] M. Strous, B. Kraft, R. Bisdorf, and H. Tegetmeyer, "The binning of metagenomic contigs for microbial physiology of mixed cultures," *Frontiers in microbiology*, vol. 3, p. 410, 2012.

[214] N. Segata, D. Börnigen, X. C. Morgan, and C. Huttenhower, "PhyloPhlAn is a new method for improved phylogenetic and taxonomic placement of microbes," *Nature communications*, vol. 4, p. 2304, 2013.

[215] J. Alneberg, B. S. Bjarnason, I. De Bruijn, M. Schirmer, J. Quick, U. Z. Ijaz, L. Lahti, N. J. Loman, A. F. Andersson, and C. Quince, "Binning metagenomic contigs by coverage and composition," *Nature methods*, vol. 11, no. 11, p. 1144, 2014.

[216] C. Luo, R. Knight, H. Siljander, M. Knip, and D. Xavier, R. J. & Gevers, "ConStrains identifies microbial strains in metagenomic datasets," *Nature Biotechnology*, vol. 33, p. 1045–105, 2015.

[217] B. Langmead and S. Salzberg, "Fast gapped-read alignment with Bowtie2," *Nature Methods*, vol. 9, pp. 357–359, 2012.

[218] D. T. Truong, A. Tett, E. Pasolli, C. Huttenhower, and N. Segata, "Microbial strain-level population structure and genetic diversity from metagenomes," *Genome research*, vol. 27, no. 4, pp. 626–638, 2017.

[219] D. T. Truong, E. A. Franzosa, T. L. Tickle, M. Scholz, G. Weingart, E. Pasolli, A. Tett, C. Huttenhower, and N. Segata, "MetaPhlAn2 for enhanced metagenomic taxonomic profiling," *Nature methods*, vol. 12, no. 10, p. 902, 2015.

[220] R. C. Edgar, "MUSCLE: a multiple sequence alignment method with reduced time and space complexity," *BMC Bioinformatics*, vol. 5, no. 1, p. 113, 2004.

[221] M. Ott, J. Zola, A. Stamatakis, and S. Aluru, "Large-scale maximum likelihood-based phylogenetic analysis on the IBM BlueGene/L," in *Proceedings of the 2007 ACM/IEEE conference on Supercomputing*. ACM, 2007, p. 4.

[222] D. Li, C.-M. Liu, R. Luo, K. Sadakane, and T.-W. Lam, "MEGAHIT: an ultra-fast single-node solution for large and complex metagenomics assembly via succinct de Bruijn graph," *Bioinformatics*, vol. 31, no. 10, pp. 1674–1676, 2015. [Online]. Available: +http://dx.doi.org/10.1093/bioinformatics/btv033

[223] H. Li and R. Durbin, "Fast and accurate short read alignment with Burrows–Wheeler transform," *Bioinformatics*, vol. 25, no. 14, pp. 1754–1760, 2009.

[224] R. L. Tatusov, M. Y. Galperin, D. A. Natale, and E. V. Koonin, "The COG database: a tool for genome-scale analysis of protein functions and evolution," *Nucleic acids research*, vol. 28, no. 1, pp. 33–36, 2000.

[225] F. D. Ciccarelli, T. Doerks, C. Von Mering, C. J. Creevey, B. Snel, and P. Bork, "Toward automatic reconstruction of a highly resolved tree of life," *Science*, vol. 311, no. 5765, pp. 1283–1287, 2006.

[226] A. Töpfer, O. Zagordi, S. Prabhakaran, V. Roth, E. Halperin, and N. Beerenwinkel, "Probabilistic inference of viral quasispecies subject to recombination," *Journal of Computational Biology*, vol. 20, no. 2, pp. 113–123, 2013.

[227] D. Jayasundara, I. Saeed, S. Maheswararajah, B. C. Chang, S.-L. Tang, and S. K. Halgamuge, "ViQuaS: an improved reconstruction pipeline for viral quasispecies spectra generated by next-generation sequencing," *Bioinformatics*, vol. 31(6), pp. 886–96, 2015.

[228] S. Prabhakaran, M. Rey, O. Zagordi, N. Beerenwinkel, and V. Roth, "HIV haplotype inference using a propagating dirichlet process mixture model," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, pp. 182–191, 2013.

[229] L. McNally and S. P. Brown, "Building the microbiome in health and disease: niche construction and social conflict in bacteria," *Phil. Trans. R. Soc. B*, vol. 370, no. 1675, p. 20140298, 2015.

[230] P. Wilmes, S. L. Simmons, V. J. Denef, and J. F. Banfield, "The dynamic genetic repertoire of microbial communities," *FEMS microbiology reviews*, vol. 33, no. 1, pp. 109–132, 2008.

[231] D. R. Zerbino and E. Birney, "Velvet: algorithms for de novo short read assembly using de Bruijn graphs," *Genome Research*, vol. 18, pp. 821–829, 2008.

[232] Li, H. and Handsaker, B. and Wysoker, A. and Fennell, T. and Ruan, J. and Homer, N. and Marth, G. and Abecasis, G. and Durbin, R. and 1000 Genome Project Data Processing Subgroup, "The sequence alignment/map (SAM) format and SAMtools," *Bioinformatics*, vol. 25, pp. 2078–9, 2009.

[233] M. DePristo, *et al.*, "A framework for variation discovery and genotyping using next-generation DNA sequencing data," *Nature Genetics*, vol. 43, pp. 491–498, 2011.

[234] T. Namiki, T. Hachiya, H. Tanaka, and Y. Sakakibara, "MetaVelvet: an extension of velvet assembler to de novo metagenome assembly from short sequence reads," *Nucleic acids research*, vol. 40, no. 20, pp. e155–e155, 2012.

[235] T. Seemann, "Prokka: rapid prokaryotic genome annotation," *Bioinformatics*, vol. 30, no. 14, pp. 2068–2069, 2014.

[236] J. Fowkes and C. Sutton, "A bayesian network model for interesting itemsets," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2016, pp. 410–425.

[237] T. Peng, I. Harris, and Y. Sawa, "Detecting phishing attacks using natural language processing and machine learning," in *Semantic Computing (ICSC), 2018 IEEE 12th International Conference on*. IEEE, 2018, pp. 300–301.

[238] S. M. Nicholls, W. Aubrey, K. De Grave, L. Schietgat, C. J. Creevey, and A. Clare, "Advances in the recovery of haplotypes from the metagenome," *bioRxiv*, p. 067215, 2016.

[239] A. Rambaut and N. C. Grass, "Seq-Gen: an application for the Monte Carlo simulation of DNA sequence evolution along phylogenetic trees," *Computer applications in the biosciences : CABIOS*, vol. 13, no. 3, pp. 235–238, 1997. [Online]. Available: http://bioinformatics.oxfordjournals.org/content/13/3/235.abstract

[240] M. Sharma and P. M. Chauhan, "Dihydrofolate reductase as a therapeutic target for infectious diseases: opportunities and challenges," *Future Medicinal Chemistry*, vol. 4, no. 10, pp. 1335–1365, 2012.

[241] F. D. Giallonardo, *et al.*, "Full-length haplotype reconstruction to infer the structure of heterogeneous virus populations," *Nucleic Acids Res*, vol. 42, no. 14, p. e115, 2014.

[242] B. Korber, B. T. Foley, C. Kuiken, S. K. Pillai, J. G. Sodroski, *et al.*, "Numbering positions in HIV relative to HXB2CG," *Human retroviruses and AIDS*, vol. 3, pp. 102–111, 1998.

[243] J. Cuevas, R. Geller, R. Garijo, J. López-Aldeguer, and R. Sanjuán, "Extremely high mutation rate of HIV-1 in vivo," *PLoS Biol*, vol. 13(9), p. e1002251, 2015.

[244] Human Microbiome Project Consortium and others, "Structure, function and diversity of the healthy human microbiome," *Nature*, vol. 486, no. 7402, pp. 207–214, 2012.

[245] S. D. Reid, C. J. Herbelin, A. C. Bumbaugh, R. K. Selander, and T. S. Whittam, "Parallel evolution of virulence in pathogenic Escherichia coli," *Nature*, vol. 406, no. 6791, p. 64, 2000.

[246] D. C. Richter, F. Ott, A. F. Auch, R. Schmid, and D. H. Huson, "MetaSim—a sequencing simulator for genomics and metagenomics," *PloS one*, vol. 3, no. 10, p. e3373, 2008.

[247] J. L. Morgan, A. E. Darling, and J. A. Eisen, "Metagenomic sequencing of an in vitro-simulated microbial community," *PloS one*, vol. 5, no. 4, p. e10209, 2010.

[248] M. R. Crusoe, *et al.*, "The khmer software package: enabling efficient nucleotide sequence analysis," 08 2015. [Online]. Available: http://dx.doi.org/10.12688/f1000research.6924.1

[249] UniProt Consortium, "UniProt: the universal protein knowledgebase," *Nucleic acids research*, vol. 45, no. D1, pp. D158–D169, 2016.

[250] M. C. Riley, W. Aubrey, M. Young, and A. Clare, "PD5: A General Purpose Library for Primer Design Software," *PLOS ONE*, vol. 8, no. 11, pp. 1–8, 11 2013. [Online]. Available: https://doi.org/10.1371/journal.pone.0080156

[251] D. T. Schultz, "pauvre: a plotting package designed for nanopore and pacbio long reads," https://github.com/conchoecia/pauvre), 2017.

[252] H. Li, "Minimap2: fast pairwise alignment for long dna sequences," *arXiv preprint arXiv:1708.01492*, 2017.

[253] C. J. Meehan and R. G. Beiko, "A phylogenomic view of ecological specialization in the Lachnospiraceae, a family of digestive tract-associated bacteria," *Genome biology and evolution*, vol. 6, no. 3, pp. 703–713, 2014.

[254] M. Krzywinski, J. Schein, I. Birol, J. Connors, R. Gascoyne, D. Horsman, S. J. Jones, and M. A. Marra, "Circos: an information aesthetic for comparative genomics," *Genome research*, vol. 19, no. 9, pp. 1639–1645, 2009.

[255] W. Shi, *et al.*, "Methane yield phenotypes linked to differential gene expression in the sheep rumen microbiome," *Genome research*, vol. 24, no. 9, pp. 1517–1525, 2014.

[256] J. Kamke, *et al.*, "Rumen metagenome and metatranscriptome analyses of low methane yield sheep reveals a sharpea-enriched microbiome characterised by lactic acid formation and utilisation," *Microbiome*, vol. 4, no. 1, p. 56, 2016.

[257] T. J. Wilkinson, S. A. Huws, J. E. Edwards, A. Kingston-Smith, K. Siu Ting, M. Hughes, F. Rubino, M. Friedersdorff, and C. Creevey, "CowPI: A rumen microbiome focussed version of the PICRUSt functional inference software," *Frontiers in Microbiology*, vol. 9, p. 1095, 2018.

[258] W.-H. Li, "Unbiased estimation of the rates of synonymous and nonsynonymous substitution," *Journal of molecular evolution*, vol. 36, no. 1, pp. 96–99, 1993.

[259] C. Creevey and J. O. McInerney, "CRANN: detecting adaptive evolution in protein-coding DNA sequences," *Bioinformatics*, vol. 19, no. 13, pp. 1726–1726, 2003.

[260] J. Huerta-Cepas, K. Forslund, L. P. Coelho, D. Szklarczyk, L. J. Jensen, C. von Mering, and P. Bork, "Fast genome-wide functional annotation through orthology assignment by eggNOG-mapper," *Molecular biology and evolution*, vol. 34, no. 8, pp. 2115–2122, 2017.

[261] R. L. Tatusov, E. V. Koonin, and D. J. Lipman, "A genomic perspective on protein families," *Science*, vol. 278, no. 5338, pp. 631–637, 1997.

[262] R. L. Tatusov, *et al.*, "The COG database: an updated version includes eukaryotes," *BMC bioinformatics*, vol. 4, no. 1, p. 41, 2003.

[263] E. Rosenberg, "The Family Prevotellaceae," in *The prokaryotes*. Springer, 2014, pp. 825–827.

[264] E. Takayuki, *Ruminococcus*. American Cancer Society, 2015, pp. 1–5. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/9781118960608.gbm00678

[265] M. J. Coyne, N. L. Zitomersky, A. M. McGuire, A. M. Earl, and L. E. Comstock, "Evidence of extensive DNA transfer between bacteroidales species within the human gut," *MBio*, vol. 5, no. 3, pp. e01 305–14, 2014.

[266] H. Daniel, *et al.*, "High-fat diet alters gut microbiota physiology in mice," *The ISME journal*, vol. 8, no. 2, p. 295, 2014.

[267] D. A. Antonopoulos, W. M. Russell, and B. A. White, "Phylogenetic reconstruction of Gram-positive organisms based on comparative sequence analysis of molecular chaperones from the ruminal microorganism Ruminococcus flavefaciens FD-1," *FEMS microbiology letters*, vol. 227, no. 1, pp. 1–7, 2003.

[268] A. El Kaoutari, F. Armougom, J. I. Gordon, D. Raoult, and B. Henrissat, "The abundance and variety of carbohydrate-active enzymes in the human gut microbiota," *Nature reviews Microbiology*, vol. 11, no. 7, p. 497, 2013.

[269] L. Wang, A. Hatem, U. V. Catalyurek, M. Morrison, and Z. Yu, "Metagenomic insights into the carbohydrate-active enzymes carried by the microorganisms adhering to solid digesta in the rumen of cows," *PloS one*, vol. 8, no. 11, p. e78507, 2013.

[270] K. E. Burnum, S. J. Callister, C. D. Nicora, S. O. Purvine, P. Hugenholtz, F. Warnecke, R. H. Scheffrahn, R. D. Smith, and M. S. Lipton, "Proteome insights into the symbiotic relationship between a captive colony of nasutitermes corniger and its hindgut microbiome," *The ISME journal*, vol. 5, no. 1, p. 161, 2011.

[271] E. C. Martens, *et al.*, "Recognition and degradation of plant cell wall polysaccharides by two human gut symbionts," *PLoS biology*, vol. 9, no. 12, p. e1001221, 2011.

[272] D. Dodd, R. I. Mackie, and I. K. Cann, "Xylan degradation, a metabolic property shared by rumen and human colonic Bacteroidetes," *Molecular microbiology*, vol. 79, no. 2, pp. 292–304, 2011.

[273] M. Zhang, J. R. Chekan, D. Dodd, P.-Y. Hong, L. Radlinski, V. Revindran, S. K. Nair, R. I. Mackie, and I. Cann, "Xylan utilization in human gut commensal bacteria is orchestrated by unique modular organization of polysaccharide-degrading enzymes," *Proceedings of the National Academy of Sciences*, vol. 111, no. 35, pp. E3708–E3717, 2014.

[274] M. Till, D. Goldstone, G. Card, G. T. Attwood, C. D. Moon, and V. L. Arcus, "Structural analysis of the GH43 enzyme Xsa43E from Butyrivibrio proteoclasticus," *Acta Crystallographica Section F: Structural Biology Communications*, vol. 70, no. 9, pp. 1193–1198, 2014.

[275] K. Břinda, V. Boeva, and G. Kucherov, "OCOCO: the first online consensus caller," https://github.com/karel-brinda/ococo, 2016.

[276] D. Aguiar, "HapCompass manual," Brown University, Tech. Rep., 2014.

[277] H. Akaike, "Likelihood of a model and information criteria," *Journal of Econometrics*, vol. 16, no. 1, pp. 3–14, 1981.

[278] A. Charuvaka and H. Rangwala, "Evaluation of short read metagenomic assembly," in *BMC genomics*, vol. 12, no. 2. BioMed Central, 2011, p. S8.

[279] J. O. McInerney and D. H. Erwin, "The role of public goods in planetary evolution," *Phil. Trans. R. Soc. A*, vol. 375, no. 2109, p. 20160359, 2017.

[280] J. F. Banfield and M. Young, "Variety—the splice of life—in microbial communities," *Science*, vol. 326, no. 5957, pp. 1198–1199, 2009.

[281] G. K. Paterson, *et al.*, "Capturing the cloud of diversity reveals complexity and heterogeneity of mrsa carriage, infection and transmission," *Nature Communications*, vol. 6, p. 6560, 2015.

[282] J. L. Gardy and N. J. Loman, "Towards a genomics-informed, real-time, global pathogen surveillance system," *Nature Reviews Genetics*, vol. 19, no. 1, p. 9, 2018.

[283] K. Zaremba-Niedzwiedzka, *et al.*, "Asgard archaea illuminate the origin of eukaryotic cellular complexity," *Nature*, vol. 541, no. 7637, p. 353, 2017.

[284] J. O. McInerney and M. J. O'connell, "Microbiology: mind the gaps in cellular evolution," *Nature*, vol. 541, no. 7637, p. 297, 2017.

[285] R. Wallace, K.-J. Cheng, D. Dinsdale, and E. Ørskov, "An independent microbial flora of the epithelium and its role in the ecomicrobiology of the rumen," *Nature*, vol. 279, no. 5712, p. 424, 1979.

[286] J. P. Brooks, *et al.*, "The truth about metagenomics: quantifying and counteracting bias in 16S rRNA studies," *BMC microbiology*, vol. 15, no. 1, p. 66, 2015.

[287] G. Henderson, F. Cox, S. Kittelmann, V. H. Miri, M. Zethof, S. J. Noel, G. C. Waghorn, and P. H. Janssen, "Effect of DNA extraction methods and sampling techniques on the apparent structure of cow and sheep rumen microbial communities," *PLoS One*, vol. 8, no. 9, p. e74787, 2013.

[288] S. Bag, *et al.*, "An improved method for high quality metagenomics DNA extraction from human and environmental samples," *Scientific reports*, vol. 6, p. 26775, 2016.

[289] Y. Zou, M. G. Mason, Y. Wang, E. Wee, C. Turni, P. J. Blackall, M. Trau, and J. R. Botella, "Nucleic acid purification from plants, animals and microbes in under 30 seconds," *PLoS biology*, vol. 15, no. 11, p. e2003916, 2017.

[290] J. Tremblay, K. Singh, A. Fern, E. S. Kirton, S. He, T. Woyke, J. Lee, F. Chen, J. L. Dangl, and S. G. Tringe, "Primer and platform effects on 16S rRNA tag sequencing," *Frontiers in microbiology*, vol. 6, p. 771, 2015.

[291] S. M. Karst, M. S. Dueholm, S. J. McIlroy, R. H. Kirkegaard, P. H. Nielsen, and M. Albertsen, "Thousands of primer-free, high-quality, full-length SSU rRNA sequences from all domains of life," *bioRxiv*, 2016. [Online]. Available: https://www.biorxiv.org/content/early/2016/08/22/070771

[292] M. A. Quail, M. Smith, P. Coupland, T. D. Otto, S. R. Harris, T. R. Connor, A. Bertoni, H. P. Swerdlow, and Y. Gu, "A tale of three next generation sequencing platforms: comparison of Ion Torrent, Pacific Biosciences and Illumina MiSeq sequencers," *BMC genomics*, vol. 13, no. 1, p. 341, 2012.

[293] I. Kozarewa, Z. Ning, M. A. Quail, M. J. Sanders, M. Berriman, and D. J. Turner, "Amplification-free Illumina sequencing-library preparation facilitates improved mapping and assembly of (G+ C)-biased genomes," *Nature methods*, vol. 6, no. 4, p. 291, 2009.

[294] M. G. Ross, C. Russ, M. Costello, A. Hollinger, N. J. Lennon, R. Hegarty, C. Nusbaum, and D. B. Jaffe, "Characterizing and measuring bias in sequence data," *Genome biology*, vol. 14, no. 5, p. R51, 2013.

[295] F. Sanger, S. Nicklen, and A. R. Coulson, "DNA sequencing with chain-terminating inhibitors," *Proceedings of the national academy of sciences*, vol. 74, no. 12, pp. 5463–5467, 1977.

[296] J. Shendure and H. Ji, "Next-generation DNA sequencing," *Nature biotechnology*, vol. 26, no. 10, p. 1135, 2008.

[297] S. Bennett, "Solexa ltd," *Pharmacogenomics*, vol. 5, no. 4, pp. 433–438, 2004.

[298] D. R. Bentley, *et al.*, "Accurate whole human genome sequencing using reversible terminator chemistry," *nature*, vol. 456, no. 7218, p. 53, 2008.

[299] S. Goodwin, J. D. McPherson, and W. R. McCombie, "Coming of age: ten years of next-generation sequencing technologies," *Nature Reviews Genetics*, vol. 17, no. 6, p. 333, 2016.

[300] D. R. Bentley, "Whole-genome re-sequencing," *Current opinion in genetics & development*, vol. 16, no. 6, pp. 545–552, 2006.

[301] E. R. Mardis, "A decade's perspective on dna sequencing technology," *Nature*, vol. 470, no. 7333, p. 198, 2011.

[302] A. E. Minoche, J. C. Dohm, and H. Himmelbauer, "Evaluation of genomic high-throughput sequencing data generated on Illumina HiSeq and genome analyzer systems," *Genome biology*, vol. 12, no. 11, p. R112, 2011.

[303] Y. Kodama, M. Shumway, and R. Leinonen, "The Sequence Read Archive: explosive growth of sequencing data," *Nucleic acids research*, vol. 40, no. D1, pp. D54–D56, 2011.

[304] J. Eid, *et al.*, "Real-time dna sequencing from single polymerase molecules," *Science*, vol. 323, no. 5910, pp. 133–138, 2009.

[305] H. Lu, F. Giordano, and Z. Ning, "Oxford Nanopore MinION sequencing and genome assembly," *Genomics, proteomics & bioinformatics*, vol. 14, no. 5, pp. 265–279, 2016.

[306] D. A. Rasko, *et al.*, "Origins of the E. coli strain causing an outbreak of hemolytic–uremic syndrome in Germany," *New England Journal of Medicine*, vol. 365, no. 8, pp. 709–717, 2011.

[307] S. Koren, G. P. Harhay, T. P. Smith, J. L. Bono, D. M. Harhay, S. D. Mcvey, D. Radune, N. H. Bergman, and A. M. Phillippy, "Reducing assembly complexity of microbial genomes with single-molecule sequencing," *Genome biology*, vol. 14, no. 9, p. R101, 2013.

[308] D. Sharon, H. Tilgner, F. Grubert, and M. Snyder, "A single-molecule long-read survey of the human transcriptome," *Nature biotechnology*, vol. 31, no. 11, p. 1009, 2013.

[309] N. J. Loman, C. Constantinidou, J. Z. Chan, M. Halachev, M. Sergeant, C. W. Penn, E. R. Robinson, and M. J. Pallen, "High-throughput bacterial genome sequencing: an embarrassment of choice, a world of opportunity," *Nature Reviews Microbiology*, vol. 10, no. 9, p. 599, 2012.

[310] R. A. White III, S. J. Callister, R. J. Moore, E. S. Baker, and J. K. Jansson, "The past, present and future of microbiome analyses," *Nature Protocols*, vol. 11, no. 11, p. 2049, 2016.

[311] M. Jain, H. E. Olsen, B. Paten, and M. Akeson, "The Oxford Nanopore MinION: delivery of nanopore sequencing to the genomics community," *Genome biology*, vol. 17, no. 1, p. 239, 2016.

[312] D. Deamer, M. Akeson, and D. Branton, "Three decades of nanopore sequencing," *Nature biotechnology*, vol. 34, no. 5, p. 518, 2016.

[313] G. Menestrina, "Ionic channels formed by Staphylococcus aureus alpha-toxin: Voltage-dependent inhibition by divalent and trivalent cations," *The Journal of membrane biology*, vol. 90, no. 2, pp. 177–190, 1986.

[314] S. Garaj, W. Hubbard, A. Reina, J. Kong, D. Branton, and J. Golovchenko, "Graphene as a subnanometre trans-electrode membrane," *Nature*, vol. 467, no. 7312, p. 190, 2010.

[315] D. Stoddart, A. J. Heron, E. Mikhailova, G. Maglia, and H. Bayley, "Single-nucleotide discrimination in immobilized DNA oligonucleotides with a biological nanopore," *Proceedings of the National Academy of Sciences*, vol. 106, no. 19, pp. 7702–7707, 2009.

[316] T. Laver, J. Harrison, P. O'neill, K. Moore, A. Farbos, K. Paszkiewicz, and D. J. Studholme, "Assessing the performance of the Oxford Nanopore Technologies MinION," *Biomolecular detection and quantification*, vol. 3, pp. 1–8, 2015.

[317] N. J. Loman, J. Quick, and J. T. Simpson, "A complete bacterial genome assembled de novo using only nanopore sequencing data," *Nature methods*, vol. 12, no. 8, p. 733, 2015.

[318] A. S. Mikheyev and M. M. Tin, "A first look at the Oxford Nanopore MinION sequencer," *Molecular ecology resources*, vol. 14, no. 6, pp. 1097–1102, 2014.

[319] M. Pallen, "Diagnostic metagenomics: potential applications to bacterial, viral and parasitic infections," *Parasitology*, vol. 141, no. 14, pp. 1856–1862, 2014.

[320] J. Quick, *et al.*, "Rapid draft sequencing and real-time nanopore sequencing in a hospital outbreak of Salmonella," *Genome Biology*, vol. 16, no. 1, p. 114, 2015.

[321] J. Quick, *et al.*, "Real-time, portable genome sequencing for Ebola surveillance," *Nature*, vol. 530, no. 7589, p. 228, 2016.

[322] A. Edwards, A. R. Debbonaire, B. Sattler, L. A. Mur, and A. J. Hodson, "Extreme metagenomics using nanopore DNA sequencing: a field report from Svalbard, 78 N," *bioRxiv*, 2016. [Online]. Available: https://www.biorxiv.org/content/early/2016/09/07/073965

[323] A. A. Votintseva, *et al.*, "Same-day diagnostic and surveillance data for tuberculosis via whole-genome sequencing of direct respiratory samples," *Journal of clinical microbiology*, vol. 55, no. 5, pp. 1285–1298, 2017.

[324] P. Bradley, H. den Bakker, E. Rocha, G. McVean, and Z. Iqbal, "Real-time search of all bacterial and viral genomic data," *bioRxiv*, 2017. [Online]. Available: https://www.biorxiv.org/content/early/2017/12/18/234955

[325] D. J. Lipman and W. R. Pearson, "Rapid and sensitive protein similarity searches," *Science*, vol. 227, no. 4693, pp. 1435–1441, 1985.

[326] W. R. Pearson and D. J. Lipman, "Improved tools for biological sequence comparison," *Proceedings of the National Academy of Sciences*, vol. 85, no. 8, pp. 2444–2448, 1988.

[327] P. J. Cock, C. J. Fields, N. Goto, M. L. Heuer, and P. M. Rice, "The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants," *Nucleic acids research*, vol. 38, no. 6, pp. 1767–1771, 2009.

[328] G. A. for Genomics and Health, "The Variant Call Format (VCF) Version 4.2 Specification," https://samtools.github.io/hts-specs/VCFv4.2.pdf, 2017.

[329] P. Compeau and P. Pevzner, *Bioinformatics algorithms: an active learning approach*. Active Learning Publishers, 2015.

[330] P. Medvedev, K. Georgiou, G. Myers, and M. Brudno, "Computability of models for sequence assembly," in *Algorithms in Bioinformatics*. Springer, 2007, pp. 289–301.

[331] M. Pop, "Genome assembly reborn: recent computational challenges," *Briefings in bioinformatics*, vol. 10, no. 4, pp. 354–366, 2009.

[332] V. Kunin, A. Copeland, A. Lapidus, K. Mavromatis, and P. Hugenholtz, "A bioinformatician's guide to metagenomics." *Microbiology and molecular biology reviews : MMBR*, vol. 72, no. 4, pp. 557–78, Table of Contents, Dec. 2008.

[333] M. Pop, "Genome assembly reborn: recent computational challenges." *Briefings in bioinformatics*, vol. 10, no. 4, pp. 354–66, July 2009.

[334] Y. Peng, H. C. Leung, S.-M. Yiu, and F. Y. Chin, "Meta-IDBA: a de novo assembler for metagenomic data," *Bioinformatics*, vol. 27, no. 13, pp. i94–i101, 2011.

[335] S. Boisvert, F. Raymond, É. Godzaridis, F. Laviolette, and J. Corbeil, "Ray Meta: scalable de novo metagenome assembly and profiling," *Genome biology*, vol. 13, no. 12, p. R122, 2012.

[336] Z. Li, *et al.*, "Comparison of the two major classes of assembly algorithms: overlap–layout–consensus and de-bruijn-graph," *Briefings in functional genomics*, vol. 11, no. 1, pp. 25–37, 2012.

[337] P. E. C. Compeau, P. a. Pevzner, and G. Tesler, "How to apply de Bruijn graphs to genome assembly." *Nature biotechnology*, vol. 29, no. 11, pp. 987–91, Nov. 2011.

[338] E. Garrison *et al.*, "vg: the variation graph toolkit," https://github.com/vgteam/vg, 2016.

[339] H. P. Eggertsson, *et al.*, "Graphtyper enables population-scale genotyping using pangenome graphs," *Nature genetics*, vol. 49, no. 11, p. 1654, 2017.

[340] H. Li, "Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM," *arXiv preprint arXiv:1303.3997*, 2013.

[341] F. Sievers and D. G. Higgins, "Clustal Omega, accurate alignment of very large numbers of sequences," in *Multiple sequence alignment methods*. Springer, 2014, pp. 105–116.

[342] N. A. O'Leary, *et al.*, "Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation," *Nucleic acids research*, vol. 44, no. D1, pp. D733–D745, 2015.

[343] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, "Basic local alignment search tool," *Journal of molecular biology*, vol. 215, no. 3, pp. 403–410, 1990.

[344] B. Buchfink, C. Xie, and D. H. Huson, "Fast and sensitive protein alignment using DIAMOND," *Nature Methods*, vol. 12, pp. 59–60, 2015.

[345] R. C. Edgar, "Search and clustering orders of magnitude faster than BLAST," *Bioinformatics*, vol. 26, no. 19, pp. 2460–2461, 2010.

[346] International SNP Map Working Group and others, "A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms," *Nature*, vol. 409, no. 6822, p. 928, 2001.

[347] A. Chakravarti, "Single nucleotide polymorphisms:... to a future of genetic medicine," *Nature*, vol. 409, no. 6822, p. 822, 2001.

[348] H. Li, "A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data," *Bioinformatics*, vol. 27, no. 21, pp. 2987–2993, 2011.

[349] E. G. Zoetendal, A. von Wright, T. Vilpponen-Salmela, K. Ben-Amor, A. D. Akkermans, and W. M. de Vos, "Mucosa-associated bacteria in the human gastrointestinal tract are uniformly distributed along the colon and differ from the community recovered from feces," *Applied and environmental microbiology*, vol. 68, no. 7, pp. 3401–3407, 2002.

Appendix A

Terminology and Techniques for Metagenomic Data

In this Chapter, I will describe some of the terminology and techniques for metagenomic data analysis, presenting some of the core concepts of typical bioinformatics workflows, namely: Assembly, Alignment and Variant Calling. Although pertinent to aspiring metagenomicists and bioinformaticians, this overview aims to serve as a glossary and provide some context for the work that I present in this thesis, as opposed to a full academic review.

A.1 Biases

One should be aware of potential sources of bias for an experiment involving microbial communities:

- **Sampling Bias**
  Sampling from a microbial community itself will likely induce bias. Members of the community may exist in low abundance, or experience shifts in population over time and space. Thus species (or strains) captured by a sample may not be fully representative of the community, both in their quantitative distribution and presence. Indeed, it has been shown that the composition of the rumen undergoes successional colonization after feeding [104], and it was suggested before the 1980s that we must consider the rumen as three interacting populations, with distinct organisms thriving on the epithelial walls, in the rumen fluid itself, and associated with plant matter [285]. When designing an experiment, one must be aware (or at least open to the possibility) of subpopulations or other confounding effects introduced by time or location.

- **DNA Extraction Bias**
  The choice of DNA extraction method, and clean-up kit can drastically change the apparent diversity of a sampled microbial community [286, 287]. Different methods for lysis are more or less effective against certain types of cell wall (*i.e.* gram-positive and negative bacteria), and reliable extraction of DNA from a heterogeneous community remains a challenge [288, 289]. When targeting particular organisms or motifs, the chosen primers could bind more efficiently to some sequences, or non-specifically to incorrect regions, yielding bias in amplification [290]. Recent work has highlighted that rRNA databases central to determination of diversity in 16S taxonomic studies are "underpopulated and skewed" to particular ecosystems, biasing downstream analyses [291]. One's choice of extraction kit and primers must be well considered.

- **Sequencing Bias**
  As briefly discussed (Section A.2), different DNA sequencing chemistries each introduce some form of error, or bias. GC-content of a DNA sequence has an effect on the resulting sequencing coverage with high-throughput methods [292] and high-GC continues to pose a challenge, even for Sanger sequencing [293]. Also, artefacts introduced during library preparation can lead to duplicate sequenced reads [293]. Particular sequence motifs, repetitive regions and runs of the same base (homopolymers) all pose problems for modern sequencing technologies [294]. Appropriate sequencing quality control strategies can help to identify and correct such biases.

- **Computational Bias**
  Bioinformatics tools and workflows often feature many user-defined parameters (such as k-mer size for assembly), the selection of which will impose assumptions and constraints on results. Basecalling, QC, assembly, alignment and variant calling all influence the available evidence. The choice of sequence or feature databases, and references strongly direct comparative analysis. Available resources may dictate a need for subsampling, or other forms of input data reduction.

A.2 DNA Sequencing

In this section I will provide a brief overview of several DNA sequencing technologies in use for the study of metagenomes. Note that this is not intended as a review of all current and upcoming sequencing technologies, but introduces methods referred to and employed as part of this thesis.

A.2.1 Chain Termination Sequencing

The first efficient method for determining nucleotide sequences, developed in 1977 by Frederick Sanger *et al.*, was chain termination sequencing [295]. The method, which is now somewhat better known as "**Sanger sequencing**"[1], employs the use of dideoxynucleotides[2] (ddNTPs) that when incorporated into a chain of nucleotides, terminate the chain and prevent the addition of further nucleotides. Given a DNA template, primer pair, polymerase, mixed dNTPs and a smaller proportion of one of the four possible ddNTPs (*e.g.* ddATP), one is able to generate a significant number of fragments of varying length, each stochastically terminated at different positions during the extension phase of polymerase chain reaction (PCR). The fragment's length and its terminating ddNTP, distinguishes which base is at the position that corresponds to the fragment size.

Modern Sanger sequencing – as commercialised by Applied Biosystems – does not require one to perform the four ddNTP reactions separately, instead pooling ddNTPs labelled with fluorescent dyes that can be differentiated at particular wavelengths. Following PCR, the resulting fragments are pulled through a capillary at a speed proportional to the fragment size (capillary electrophoresis), past a laser that can read the fluorescent labels, yielding a *chromatogram* that can be interpreted to determine the nucleotide at each position of the sequence, with some confidence [178].

Sanger sequencing yields incredibly high-quality sequence, with base call accuracies as high as 99.999% [296], and recovers substantially longer sequences than shotgun sequencing methods. Although a returned sequence is long (up to 1 kbp [296], but more typically 500 - 700 bp) and of high quality, the method aims to recover just one single DNA sequence. Although variation can be observed on the raw chromatogram, one cannot determine which variants co-occur, or determine signal from background noise arising from the chemistry, precluding the use of Sanger sequencing for haplotyping. The cost to submit DNA for Sanger sequencing as part of this work was just a few GBP per sample, making chain-terminated sequencing suitable for a low cost quality check of PCR amplicons.

[1] Chain-termination sequencing earned Sanger his second Nobel Prize, shared with Gilbert and Berg in 1980.

[2] The *deoxy-* prefix indicates an *OH* group has been replaced by *H*. Here, *dideoxy-* refers to further oxygen reduction by replacement of an additional *OH*. Loss of this hydroxyl precludes a phosphodiester bond, terminating a nucleotide chain.

A.2.2 Cyclic Reversible Termination Sequencing by Synthesis

Perhaps more recognisable when referred to as "Solexa"[3] or "Illumina sequencing", cyclic reversible termination (CRT) elongates strands under synthesis in cycles, one nucleotide at a time [297]. Briefly, for library preparation, template DNA is broken into short fragments, adapters are ligated on to both ends and the strands are separated. The library is then loaded into a "flow cell": a consumable product approximately the size of a glass microscopy slide, containing eight tubules whose walls are coated in a "lawn" of oligos complementary to one of the two end adapters. In a pre-sequencing process called bridge amplification, unbound ends of the strands also bind to the lawn, bending to creating ssDNA bridges that become double stranded via a PCR-like process. After repeated denaturing and extension steps, the initial strand will have seeded 1000 new copies, tightly packed together in a cluster whose shape is directed by the patterned flow cell wall [298, 299].

Sequencing is conducted by washing four concentrated sets of fluorescently labelled dTNPs (one for each nucleotide), polymerase and primers through the flow cell. Akin to those used in Sanger sequencing, the dNTPs are chain terminating, ensuring that only one nucleotide can be bound to the strands of each cluster [299]. The flow cell is then imaged, with each cluster of strands fluorescing to indicate the most recently appended nucleotide. Once imaged, the flow cell is washed with a reagent that breaks the bond between the fluorescing terminator and the nucleotide that is now part of the reconstructed strand, allowing the chain to be continued. The process is cycled n times, extending the strands of every cluster by n nucleotides, which are each imaged n times. The laser and camera data is converted into base calls, producing a read from each clonal cluster, and a nucleotide (with some indication of confidence) for every cycle [299, 296].

It is not an understatement to describe the initial release and later enhancement of Illumina's sequencing technology as one of the significant milestones in the history DNA sequencing; the original Solexa Genome Analyzer (2005) could offer reads of just 36 bp, but for each of the *millions* of clonal clusters occupying a flow cell [296, 300], increasing the possible throughput of a DNA sequencer by several orders of magnitude [301]. Later enhancements would lead to the availability of the Illumina HiSeq 2000 (2009), increasing read lengths to 100 - 150 bp [301].

Illumina sequencers make up the largest market share of DNA sequencing machinery [302, 299], dwarfing its competitors in terms of raw data stored in the Sequence Read Archive (SRA) [303]. Note that Illumina platforms in general are to what I referring to when I discuss "**shotgun sequencing**" or "shotgun metagenomics". According to the manufacturer's website, the newest Illumina MiSeq can generate up to 25 million reads with a maximum of 300 bp, and Illumina HiSeq 4000 can produce up to 5 billion 150 bp reads. If paired-end reads overlap with sufficient coverage, they could be used to identify the haplotypes for small protein domains, but generally they are too short to facilitate the recovery of haplotypes with CRT sequencing alone.

[3] Solexa was a spin-off venture from Cambridge University, formed in 1998. The first Solexa sequencer was built and released in 2006, the company was acquired by Illumina a year later.

A.2.3 Zero-Mode Waveguide Sequencing

In 2009, a research group at Pacific Biosciences (PacBio) published their method for real-time sequencing of single molecules, using a single polymerase enzyme to perform uninterrupted synthesis of a single template incorporating dNTPs labelled with detectable fluorophores [304]. PacBio "SMRT" (single molecule real-time) sequencing is performed in a nanoscale chamber called a *Zero-Mode Waveguide* (**ZMW**[4]). A ZMW is illuminated through its glass base which is constructed in such a way that it acts as a microscope capable of focusing on a 20 zeptolitre (zl; 10^{-21} litre) volume at the base of the ZMW, where a single DNA polymerase and template are anchored [304, 299].

The target template is prepared with a pair of hairpin adapters to coerce it to become topologically circular before it is bound to a polymerase and loaded into a ZMW via diffusion or magnetic beads. The hairpins permit continuous circular synthesis of a template (up to 3 kbp), creating a ssDNA molecule composed of multiple copies of the template, generating a *circular consensus sequence* (CCS). During sequencing, the polymerase detaches phospholinked fluorophores from dNTPs to incorporate them to the strand under synthesis, the colour and duration of light emitted as a result is captured by a laser and camera through the glass plate and translated to sequence data [299]. The glass plate offers a window into thousands of individual ZMW pores on a 1 cm^2 "SMRT cell", which is loaded into a machine that is approximately the size of a large chest freezer.

A single cell generates around 55,000 reads with an average length of 20 kbp [299] – orders of magnitude larger than what could previously be achieved with DNA sequencing. Although the error rate is significantly higher than those generated by Illumina technologies (10%-15%), this error can be amortized by inspecting the circular consensus [305]. The PacBio RS II was first made commercially available in 2013, but the newer PacBio Sequel has reduced the cost, size and weight of the SMRT sequencing technology, as well as improving the yield compared to the RS II (additional ZMW).

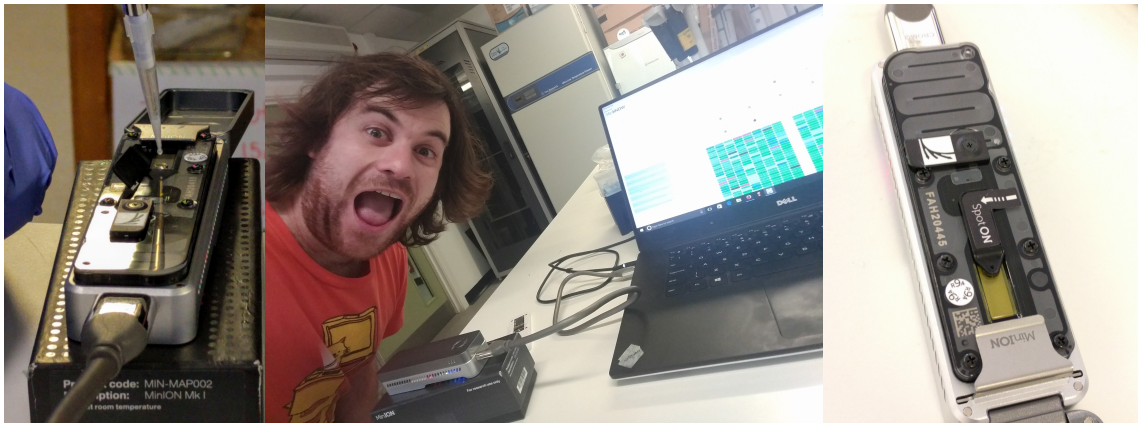
The method quickly proved its use; in 2011 SMRT was used to identify the genomic sequence of an outbreak strain of *Escherichia coli* responsible for an epidemic incident in Germany [306]. PacBio sequencing significantly reduced the cost and complexity of sequencing bacterial genomes, bringing the capability to increase the number and quality of references for population-scale studies [307]. Although recent work has shown that it is indeed possible to sequence entire RNA molecules with SMRT and conduct a small survey of a human transcriptome [308]; both the effort of library preparation [309] and the throughput of the technology [310] obstruct large-scale analysis of a metagenome, such as the haplotyping of individual isoforms.

[4] A complex name for a microscopic tube of aluminium.

A.2.4 Nanopore Strand Sequencing

The commercial release of the Oxford Nanopore Technologies' (ONT) `MinION` in 2014 [311] is a realisation of almost three decades of work[5] toward the development of nanoscale pores capable of detecting the passage of single-stranded DNA (ssDNA) and converting the changes in voltage signal to nucleotide sequence [312]. The `MinION` is a pocket-sized, portable DNA sequencer[6] (Figure A.1) that can be operated with an off-the-shelf laptop or desktop computer. The consumable "flow cell" contains an array of 2048 1 nm diameter biological[7] nanopores based on a naturally occurring protein[8] that are controlled in groups of four (permitting 512 pores to be reporting voltages simultaneously) with a bespoke microchip (an ASIC: *application-specific integrated circuit*).

Each pore generates real-time electrical signals (processed by the ASIC) as a strand of DNA passes through it (caused by nucleotides altering ion flow within the pore). Combinations of k bases (k-mers) have a signature, allowing the voltage signal "squiggles" to be converted into the molecule's DNA



(a) Careful loading of prepared library

(b) `MinION` in use, reporting real-time status to connected laptop (excited person for scale)

(c) A `MinION` flow cell

Figure A.1 My Oxford Nanopore Technologies `MinION` DNA sequencer, used for verification of my metagenomic haplotyping framework in Chapter 5

[5] David Deamer's sketch of DNA "driven through a small channel", with each base causing "a change in the current", hypothesising that this will allow one to determine the nucleotide's identity is dated on a Sunday in June 1989 [312]

[6] "Measuring $10 \times 3 \times 2$ cm and weighing just 90 g" [305]

[7] Biological nanopores actually exist in nature. *Staphylococcus aureus* secretes α-hemolysin (α-toxin) proteins as a toxin [313], but we have exploited it for strand sequencing! Though, ONTs website (accessed Spring 2018) describes future work directed to the development of solid-state mechanical nanopores, including the possibility of using graphene [314].

[8] The pores are genetically modified to reduce the size of α-hemolysin's sensing region, as about 12 nucleobases could be contained at once in its 5 nm stem, obscuring identification of specific bases [312]. This was achieved by Stoddard *et al.* [315]. Hagan Bayley, the corresponding author of the work, would go on to found ONT in 2005.

sequence. The distribution of read lengths is similar to that of Pacific Biosciences, though with careful preparation one can achieve single reads over a megabase[9] in length[10].

However, reported base calling accuracy ranges between 65% to 88% [312] and error is effectively randomly distributed as noise, making it difficult to identify and correct read error systematically; requiring more computationally expensive overlap-based methods for assembly (Section A.4) [305]. My own experiment indicates the `MinION` has an error rate of approximately 10% (Section 5.3). Additionally, the platform is particularly prone to error arising from repeated runs of the same nucleotide (homopolymers) [194]. Despite the error rate and profile, long reads yield more contiguous assembly [316]. Indeed, it has been possible to assemble complete consensus bacterial [317] and human [146] genomes with nanopore strand sequencing. However, high DNA input requirements [318] and the rate of errors currently precludes the use of nanopore strand sequencing for the direct recovery of individual haplotypes from a metagenome.

Owing to its portability, rapid results and price[11], the `MinION` has found itself as an attractive option, particularly for real-time pathogen surveillance [319, 320] and sequencing in remote locations [321, 322]. Recent work has demonstrated same day *Mycobacterium tuberculosis* [323] identification. Fast results, combined with rapid screening [324], holds powerful potential for future clinical diagnostics.

The technology underlying the `MinION` can be scaled up with hardware capable of running multiple flow cells simultaneously (`GridION`; 5 flow cells, `PromethION`; 48 flow cells).

[9] Adam Philippy (NHGRI) introduced an oft-used analogy in a recent conference talk: if a nanopore was the size of a fist, a 1 Mbp strand of DNA passing through the pore would be 3.2 km long, and move at almost a metre per second

[10] As of February 2018, the longest reported Nanopore read mapping is 1.3 Mbp, reported by Matt Loose, claiming "the Nanopore Ashes" trophy for the longest read (for now...): https://twitter.com/mattloose/status/954147458778587136

[11] Our starter pack was priced at around £1000 and included the `MinION` itself, 2 flowcells and sample preparation kit

A.3 Data Handling

A.3.1 Common File Formats

I will provide a brief overview of some common file formats that are encountered in the day-to-day work of a bioinformatician. These formats will be referred to without expansion in this thesis.

FASTA

FASTA (pronounced "Faster" or "Fast A") is a plaintext format for storing nucleotide or peptide sequences. The format was created in 1985, to specify input data to Lipman and Pearson's protein similarity search program `FASTP` [325] (a forerunner to `BLAST`). The format is occasionally confused with the `FASTA` program, developed in 1988 as a follow up to `FASTP` that permitted DNA to DNA and translated protein to DNA searches [326].

The format is loosely defined, with no official specification. `FASTA` files require a sequence description, delimited by '>', followed by the DNA or amino acid sequence itself on one or more lines. The file is permitted to contain more than one record. Listing A.1 provides an example.

```
>Example_Sequence_1
TAGCGATTTATCGGAGCGCCTCGGAATACGGTATGAGCAGGCGCCTCGTGAGACCATTGCGAATACCAGGTATCGTGTA
AGTAGCGAAGGCCCGTACGCGAGATAAACTGCTAGGAAACCGCGTCTCTACGACCGGTGCTCGATTTAATTTCGCTGAC
>Example_Sequence_2
CGGGGTACTCTTGCTATCCATATGGTCCACAGGACACTCGTTGTTTTCGGATTTACCCTTTATGCGCCGGTTTTCAGCC
ACGCTTATGCCCAGCATCGTTACAACCAGACCGATACTAGATGTATAAAGTCCGCCATGCAGACGAGACCAGTCGGAGA
```

Listing A.1 An example `FASTA` file consisting of two DNA sequence records.

A ';' may be interpreted as a comment, typically ignored by the majority of software designed to handle `FASTA` files, though it is rarely used in practice. Additionally, an unofficial *de facto* standard for the formatting of the sequence description lines was introduced by the NCBI, to ease parsing accession identifiers from commonly used databases. For example, `GenBank` entries are named with the template: `>gb|accession|locus`. It is of note that there is no official file extension associated with `FASTA` data. Both `.fasta` and `.fa` are common. Occasionally, one may encounter `.fna` and `.faa` which imply that a `FASTA` contains nucleotide or amino acid sequences, respectively.

FAI

To permit fast and efficient random access to records in a FASTA, one must generate a corresponding index. A FASTA index is stored in a **FAI** file. Although there is no formal specification, the index is generated in the same tab-delimited, human-readable format by many tools designed to handle FASTA files. Almost exclusively, it is assumed that the index will have the extension .fai appended to the name of the FASTA file, and will exist in the same directory. Each sequence in a FASTA has a corresponding row in the index that contains:

- the sequence name
- the total sequence length (nt or aa)
- byte-offset of the first letter of the sequence in the FASTA
- maximum number of nucleotides or amino acids per line
- maximum number of bytes per line (usually one more than the previous field)

Note that an index assumes (and thus requires) that the nucleotide (or amino acid) lines for a given sequence are the same length (except for the last one). Listing A.2 provides an example, containing a small selection of lines from a FAI generated for the hs37d5 human genome reference.

```
1  1       249250621       52              60      61
2  2       243199373       253404903       60      61
3  [...]
4  X       155270560       2929051733      60      61
5  Y       59373566        3086910193      60      61
6  MT      16569           3147273397      70      71
```

Listing A.2 An example FASTA index.

FASTQ

The **FASTQ** ("Fast Q") format can be thought of as "FASTA with Qualities"; extending the FASTA format to include information on the per-position quality of the sequences. The FASTQ format was formally defined by Cock *et al.* in 2009 at the Wellcome Trust Sanger Institute, out of a need to clear up some confusion that had been caused in the community by a lack of ownership for the format and multiple competing variants of the format produced by Solexa and Illumina at the time [327].

A FASTQ record consists of four lines, containing:

- the sequence name, delimited by '@'
- the DNA sequence

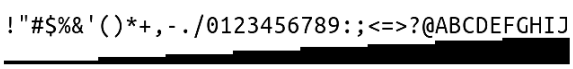


Figure A.2 A graphical depiction of how ASCII characters encode sequence quality in FASTQ files.

- the '+' character
- per-base quality scores (according to the sequencer) corresponding to the DNA sequence in the second line, encoded with single ASCII characters (see Figure A.2)

Listing A.3 provides an example. Sequences are delimited by '@', which unfortunately is also a valid quality character[12], making the parsing of FASTQ files a little precarious for new bioinformaticians. Although not required, the sequence and quality lines of a FASTQ record are usually on a single line (unlike FASTA files), primarily to avoid further complication arising from the ambiguity of '@'.

Generally[13], the ASCII characters correspond to values between 0 and 40, which correspond to the phred [178] scores for each base position. Scores range from 0 ('!'), indicating an error probability[14] of 1.0, through to 40 ('I') with an error probability of 0.0001.

Both FASTA and FASTQ are ubiquitous in bioinformatics, partly owing to their trivial, human-readable format. Despite their age, and inefficient storage requirements, FASTA and FASTQ remain *de facto* formats for containing sequence databases, and raw sequenced reads, respectively.

```
1 @HISEQ:99:C4CPTACXX:1:1106:13483:6473 1:N:0:ACAGTG
2 CCGTCCAATAATTTTAATAGCGTGACCACCTAGATATGAACCAGTAACGTGTTGGTAAACTCCACTTTTATAATTAGCA
3 AAATCTTCATATACACTGAAA
4 +
5 BB<FFBFFF<BFFFFIBBB0<B7BB<0BF7B7<B'<07B<F7BBFB<FFFFFIIIFB0'<BFB<FFIIFFF0<BFB'7<
6 BFB<'7B<<BBFBBB'<B077
```

Listing A.3 An example FASTQ file containing a single read from a real sequencing experiment.

VCF

The *Variant Call Format* (**VCF**) defines a file format for storing differences against a known reference genome, whose location is common across multiple samples. Using the format, one may reduce their storage requirements to a copy of the reference genome, and a VCF that enumerates the differences observed when comparing particular positions of the reference to their samples. Further development

[12] In fact, the '+' delimiter between the sequence and quality strings is *also* permitted to appear in the quality string... These decisions were jokingly described to me (by Peter Cock himself) as "one of the many sins in the FASTQ story": https://twitter.com/samstudio8/status/551040458923982849

[13] Though there have been several different variants of the FASTQ format, the quality alphabet is somewhat unified now.

[14] A phred score of 'Q' can be transformed into a probability via $10^{\frac{-Q}{10}}$ [178]

of the format was completed as part of the 1000 Genomes Project [187]. The format is well defined and since version 4.0, responsibility for maintaining the `VCF` standard has been taken over by the Global Alliance for Genomics and Health (GA4GH). The file extension is `.vcf`[15], and many tools expect the `VCF` to be compressed with `bgzip` and indexed for fast random access with `tabix`.

For the purpose of my own work in this thesis, the use of a `VCF` is limited to providing a list of 1-indexed positions along a contiguous region of DNA that are single nucleotide polymorphisms (SNPs), requiring use of just two of the eight data columns mandated by the format. Thus I leave detailed explanation of the format to the specification itself [328].

BED

The *Browser Extensible Display* (**BED**) format is a simple text-based, whitespace-delimited file format[16] for defining 'tracks' to be displayed on genomics viewers. The format has also garnered use as a generic means of storing and sharing co-ordinates for regions along a genome. The most basic `BED` consists of three columns describing the locations of features along a genome:

- contig or chromosome on which the feature appears
- start position of feature (0-based)
- end position of feature (1-based)

GFF

The *General Feature Format* (**GFF**) is a common file format for the storage of genome annotations. Like the `BED` format, it can additionally be used to draw tracks on a compatible genome viewer. The specification defines a tab-delimited, plain text file and consists of nine fields:

- sequence name
- source of annotation (tool or database)
- feature type
- start position of feature (1-based)
- end position of feature (1-based)
- feature score
- forward ('+') or reverse ('-') stranded
- codon frame (0, 1 or 2)
- additional user-specified key-value pairs (*e.g.* `UniProt` ID, EC number)

[15] Not to be confused with the `vCard` format for electronic business cards, which is also a `.vcf` file.
[16] Though some `BED` parsers expect spaces, or tabs specifically

It is worth noting that there are two versions of the format in common use. GFF3 attempts to address some of the shortcomings of GFF2; primarily its inability to handle three-level or deeper hierarchies of nested features. Gene prediction tools such as prokka [235] will generate their annotations in GFF format. In this thesis, regions of interest will be targeted by first filtering GFF files.

SAM, BAM, CRAM

Sequence Alignment/Map (**SAM**[17]) is arguably one of our field's most well specified and popular formats, designed to store alignments of DNA sequencer reads against a known reference. The format, and the samtools toolkit for handling and manipulating SAM files was developed by Li *et al.* at the Wellcome Trust Sanger Institute in 2009 [232].

A sequence aligner (Section A.5) will typically output its alignments in SAM format. The initial header section of a SAM describes the reference used, and keeps a history of transformations that have been performed on the alignments. A single read alignment is described by 11 mandatory fields:

- query name (sequenced read)
- a bitwise flag (*e.g.* flagging secondary or mate-pair status, strand direction, quality control failure)
- reference contig or chromosome name
- first mapped base position on the reference (1-based)
- probability that the mapping position is wrong
- The 'CIGAR[18]' string, describing the ordered set of operations (trims, insertions, deletions and matches) that coerce the given read to map to the reference
- reference contig/chromosome on which the next (mate) read appears
- position at which the next (mate) read appears
- the length of the reference covered by a mate pair (or 0, if not correctly paired)
- query DNA sequence (or '*')
- query per-base phred quality in FASTQ-like ASCII encoding (or '*')

In practice, the plaintext SAM format is rarely used. A machine-readable, binary, compressed version of the SAM format: **BAM**, is the *de facto* alignment standard.

CRAM files are essentially BAM files that have been further compressed by reducing the resolution of the attached quality data, and by storing information on bases only if they differ from the reference. The format was created at the European Bioinformatics Institute to address its growing storage requirements. Efforts to adapt popular bioinformatics tools to be fully CRAM-compatible are ongoing.

[17] Sharing the name of the file format and toolkit made samtools steering committee meetings incredibly confusing.
[18] Concise Idiosyncratic Gapped Alignment Report

A.4 Assembly

Currently, sequencing does not produce whole-genomes[19]. As discussed earlier (Section A.2), DNA sequencers produce **reads**: strings of DNA that originate from the original sample, offering a window onto the genome in question. The length of a read varies with the particular sequencing platform employed. Broadly, we may divide sequencing technologies into 'short' and 'long' reads, but both must still be **assembled** to produce a whole genome from its sequenced fragments. These reads will have sequence similarity to others, allowing them to overlap. These overlaps can be exploited computationally by *assemblers* to "line-up" and group reads together, to construct one (but likely many more) contiguous sequences of DNA known as '**contigs**', which are representative of part of the original genome.

The difficulty involved in assembly can be drastically reduced if one has a reference sequence for the organism under study. With a good reference, one can *align* their reads against the reference, removing the need to attempt to rebuild the genome from scratch. Though, reference guided assembly does have issues of its own, which will be discussed shortly (Section A.5).

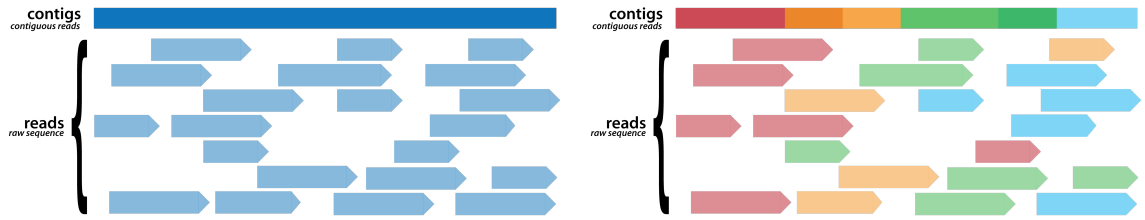
Assembly without a reference to hand is called *de novo* assembly. An oft-used metaphor[20] for the longstanding short-read *de novo* assembly problem describes the shredding of multiple copies of a book and attempting to reconstruct its contents from millions of short sub-sentence excerpts (that can also contain errors or missing words). *De novo* assembly is a computationally difficult problem (*NP-hard*, see Aside 2.1) in the field of bioinformatics [330], with development of efficient and tractable methods for assembly remaining an open research question in its own right [331]. With high-quality reference genomes still unavailable for many of the species (and certainly strains) that co-exist within communities of interest [112], many metagenomics workflows rely heavily on the construction of assemblies in a *de novo* fashion, so I will focus further discussion on *de novo* assembly only.

Unfortunately, the assembly of metagenomic samples further complicates matters, as the goal is no longer to recover a single genome, but an unknown number of genomes, each with a potentially unknown length and with unknown sequence properties. To extend our metaphor[21], we must now shred many different books, and attempt to differentiate which fragments belong to each book, with nothing other than fragment's contents as evidence. However, assemblers are designed with the goal of constructing a consensus: a DNA sequence that is representative of its input [150, 152].

[19] Though, this is an exciting time for DNA sequencing, with recent examples showing it is possible to close small genomes such as *Escherichia coli* with very few ultra-long reads: http://lab.loman.net/2017/03/09/ultrareads-for-nanopore/

[20] The exact origin for which I have been unable to track down... Though, Compeau and Pevzner formulate the metaphor as one where stacks of newspapers are exploded into confetti [329]

[21] I once had to describe this problem for a poster in my first year, where I likened the issue somewhat loosely to jigsaw puzzles: [...] it's like trying to simultaneously assemble thousands of jigsaws but some of the jigsaws are heavily duplicated and some of the jigsaws hardly appear at all, a lot of the pieces are missing and quite a few pieces that really should fit together are broken. Also the jigsaws are mostly pictures of sky.



(a) Single organism assembly

(b) Metagenomic assembly

Figure A.3 A visualisation demonstrating the fundamental difference between assembly of reads from a single organism (a) and metagenomic samples (b). Sequenced reads are depicted by pointed boxes and are arbitrarily coloured in (b) to represent reads from a mixed microbial population. The contiguous DNA sequences ('contigs') constructed by the process of assembly are depicted as a contiguous block above the reads. The contig in (b) demonstrates assembly of metagenomic reads will typically yield nothing more than a **consensus** of the input.

Yet what does *representative* even mean in the context of a metagenome? Figure A.3b depicts the alignment of reads from a mixed species population against a contig assembled from a metagenome. The assembly is a consensus of its input, likely recovering a sequence that appears to be representative of multiple species (or at least its strains), but does not actually exist in nature. Species in high abundance will often dominate sequencing experiments, resulting in an assembly biased toward these species [332] and potentially little of anything else. Intra-species variation between closely related species may not be detectable at all [333]. Clearly, this poses a significant issue for recovery of species and strain level variation in a community, as variation between strains and similar species is lost [278]. To be clear, assembly of metagenomic data does not produce haplotypes.

It is important to note that there are assemblers designed specifically for metagenomic data [334, 335, 234, 222, 151], but their goal is not to specifically disentangle complex species and strain variation. The recent manuscript for `metaSPAdes` (and arguably the *de facto* choice for metagenomic assembly currently), explicitly describes its focus "on reconstructing a consensus backbone of a strain mixture", rather than the strain-level variation itself [151]. Indeed, as I will describe in my literature review (Section 2.8.1), current workflows for handling metagenomic data instead rely on lossy and naive pre-processing steps such as read binning (to group reads that likely arise from similar genomes) in an effort to side-step the significant computational requirements, or constructing chimeras.

In Chapter 3.1, I will provide the first formal definition for the problem of recovering population-level variation from a metagenome. I will demonstrate haplotype recovery using the raw sequenced reads that align to regions of interest on an assembled metagenome (*e.g.* Figure A.3b), where the assembly is used merely as a "**pseudo-reference**" to provide a co-ordinate system for the reads themselves. Through Chapters 3.2 and 3.3, I will present my implementation of an algorithm capable of leveraging the mixture of reads that align to regions of these "pseudo-references" to recover the true variation (haplotypes) that exists within a microbial community.

A.4.1 Implementations

Ahead of my review in the following Chapter, I introduce some terminology to describe different implementations for addressing the problem of genomic assembly.

Greedy

The most simple and naive method for genomic assembly is a 'greedy' one. A greedy assembler crudely joins together reads with the 'best' overlaps (for some definition of best that likely considers the length and sequence identity of the overlap), until no reads are left to join [333]. The method is intuitive but naive, as the exhaustive calculation of all overlap scores is computationally expensive, requiring a comparison of every possible read pairing.

Note that **greedy algorithms** are not limited to the problem of assembly, and refer generally to any approach that makes decisions by only considering the immediate outcome of the current choice ('local optimisation'), and not how a choice may impact future decisions ('global optimisation'). A greedy approach is not necessarily 'bad', but they are not guaranteed to offer a globally optimal (the 'best') solution, and are likely to become trapped in local minima. In the context of assembly, greedy methods perform poorly when faced with repetitive regions [333]. In modern practice, greedy assemblers are not capable of handling the size of high-throughput sequencing data sets, and have been sidelined in favour of more robust assemblers. However, exceptions to this statement exist, and as recently as 2016, greedy assembly has been proposed to solve a niche problem (Section 2.6.3).

Overlap-Layout-Consensus

Overlap-Layout-Consensus (OLC) breaks the problem of assembly into three different eponymous steps, in an attempt to overcome the primary shortcoming of greedy methods: local optimisation precluding the ability to consider the global implications of a decision. Akin to greedy methods, overlaps between all read pairs must be exhaustively enumerated before assembly can begin. Overlaps between reads are represented by an 'overlap graph', in which each read is a node and is joined by an edge to another node if that read pair are determined to share an overlap [333].

The layout step is responsible for calculating valid paths that exist in the overlap graph. Ideally, one such *Hamiltonian Path* that traverses the graph and visits each node only once will exist, but specifically finding such a path is an NP-hard problem in itself. Despite this, the strategy of first representing the overlaps as a graph allows more robust reconstructions than OLC's greedy competition. The third and final step simply traverses the 'best' path found, and "flattening" overlapping regions into a consensus sequence, whereby each base is decided on via majority rule. OLC is discussed in more detail by Pop [333] and Li *et al.* [336].

De Bruijn Graph

With the advent of large scale sequencing experiments, it became increasingly intractable to exhaustively evaluate the overlaps necessary to construct assemblies from the significant number of raw reads, necessitating an assembly solution that could scale [231]. An advance in assembly was brought about when De Bruijn graph structures (first introduced in 1946) were applied to represent overlaps on subsequences of reads. de Bruijn graph assembly involves sharding of all the reads into smaller pieces of length *k*, called *k-mers*. An 'overlap graph' wherein nodes represent specific subsequences, are connected to other k-mers for which they are a prefix or suffix, according to the reads. The representation is efficient, as it can reduce the assembly problem to one of assembling all redundant k-mers, rather than the raw reads themselves, though some resolution is lost. For the interested reader, the method has been excellently introduced elsewhere by Compeau *et al.* [337].

De Bruijn graph assemblers have become the *de facto* methodology for assembly, underpinning the majority of the current state-of-the-art assemblers, including `MetaVelvet` [234] and `metaSPAdes` [151]. However, de Bruijn assemblies are *very* sensitive to the choice of *k*, whose optimum value depends on both the genome and read properties, and is typically left to the user to optimise.

Variant Graphs

It is clear that one's choice of genomic reference will guide interpretation of sequenced DNA, strongly influencing the conclusions of any downstream analyses. Genomic assemblers are only designed to generate a consensus of their inputs, which stand only as representative sequences for the individual members of a population, thus a linear reference can encode only one haplotype (regardless of whether it actually exists, or represents chimeric reconstruction of reads), and has the potential to bias the quality of reference-guided assembly. Yet, before outputting a consensus and losing this resolution, many assemblers encode observed variation as a graph. Recognising the advantage of this, Garrison *et al.* have formally described a method for storing references graphically and have begun the ambitious project of building `vg`: the variant graph toolkit [338] for the construction and manipulation of graph-based reference sequences [138]. Work on variant graphs is somewhat in its infancy. Paten *et al.* provide an excellent survey of current projects working toward graph-based references [99].

A recent application of a variant graph is `GraphTyper` [339]: a pipeline for improving the quality of genotype calls for an individual, particularly for highly polymorphic regions. `GraphTyper` iteratively aligns sequenced short-reads against a known reference (*e.g.* `GRCh38`), determining where SNPs and indels cause alternative paths (or "bubbles"): a variant graph. The method collapses adjacent variants to significantly reduce the number and size of bubbles. For each remaining bubble, `GraphTyper` exhaustively determines the pair of paths to cross the bubble, that is most well supported by the reads (which the authors refer to as "haplotype calls"[22]): outputting the two most likely variant options, as the genotype for the corresponding reference position.

[22] One must note, that `GraphTyper` is not a haplotyper: it is not capable of joining the genotypes from different bubbles together in the right orientation (nor does it aim to), but such language confuses the meaning of haplotype further.

A.5 Alignment

A common step as part of a bioinformatics workflow is one or more forms of sequence **alignment**:

Reference Guided Assembly (Mapping)
If a reference genome for the organism under study exists, one may 'skip' *de novo* assembly and align sequenced reads directly to the reference to attempt to recover a whole genomic sequence for an individual. Aligning against a reference can help to reduce error rates (*e.g.* assisting in the navigation of repetitive regions, if the repeats are smaller than the size of a read). Repetitive regions longer than the length of a read, or hyper-variable regions still pose trouble for assembly even when a reference is present. Metagenomic typically studies do not have this option, as references are not available.

Read Sequence Alignment
One method for checking the reliability of an assembly is to map the raw sequenced reads onto the assembly and assess the distribution of coverage. As I will describe in Section 3.1.4, this also offers a convenient method to screen large numbers of reads, using the assembly as a proxy, avoiding expensive and time-consuming pairwise alignment of all reads to a database. Popular tools for efficient alignment of short-reads to a reference or assembly include `bowtie2` [217] and `bwa-mem` [340][23].

With the use of newer sequencing technologies that yield substantially longer and noisier reads becoming commonplace, a need has arisen for specialist aligners that can scale to cope with longer reads and take platform-specific error profiles into account. Currently, `minimap2` [252] is the first (and effectively *de facto*) tool for long-read sequence alignment.

Multiple Sequence Alignment (MSA)
To explore the relationship between three or more nucleotide or amino acid sequences that are believed to have a shared evolutionary history, one can conduct a multiple sequence alignment (MSA). MSA allows one to inspect the homology of sequences in a group and attempt to reconstruct their phylogeny. Examples of tools to accomplish this include `MUSCLE` [220] and `Clustal Omega` [341].

Read and Contig Database Searches
One may annotate a set of reads or contigs given an indexed database of known sequences of interest. For example one could broadly categorise taxonomy by aligning to NCBI `RefSeq` [342] or specific functions of interest with the `UniProt` [249] protein collection. Local sequence alignment has historically been dominated by the use of `BLAST` [343], though newer, significantly more efficient methods such as `Diamond` [344] and `USEARCH` [345] are slowly gaining popularity.

[23] Despite the popularity of `bwa-mem`, it has no peer-reviewed manuscript. Amusingly, Heng Li announced that he did not have time to resubmit the work to another journal following its initial rejection, as other problems "interest me more" https://sourceforge.net/p/bio-bwa/mailman/message/30894287/

A.6 Variant Calling

The identification of single nucleotide polymorphisms (SNPs) and other structural variation along a given genome is a problem known as variant calling. Given our vested interest in our own health and wellbeing, since the discovery and enumeration of almost 1.5 million SNPs on the first human reference genome in 2001 [346] with a known impact on our susceptibility to disease [347] the focus of variant calling has historically been dominated by solutions targeted at human genomes. The two 'competing' *de facto* bioinformatics toolkits that form the basis of many workflows are the `GATK` [169] and `samtools` [232] which were built for human genetics research at the Broad Institute and the Wellcome Trust Sanger Institute respectively.

Both toolkits provide mechanisms for variant calling (`GATK-HC` and `samtools mpileup`), but were built specifically for projects on human, diploid genomes. Such tools induce a diploid bias, discarding tri- or tetra-allelic positions [186], or ignoring positions that are outliers according to the Hardy-Weinberg principle as error [348]. Understandably, this bias is troubling for downstream analyses on non-diploid experiments, such as metagenomic sequencing. Indeed, recent tooling aimed specifically for the handling of metagenomic data (or other non-human areas of focus such as viruses) often attempt to perform their own variant calling with fewer assumptions instead. This will be discussed further as part of my literature review (Sections 2.7-2.8).

Appendix B

Laboratory Notebook and Protocols

B.1 Initial PCR with GoTaq

The following pages are scanned from my laboratory notebook, showing the initial amplification of `Gretel` candidates with GoTaq polymerase. These were my first successful PCR and gel runs as a bench scientist.

PCR `2017-07-28.QT3`	1×	13×	Order
GoTaq Green Buffer	5 µl	65 µl	2
dNTPs (10 mM)	2.5 µl	32.5 µl	3
GoTaq Polymerase	0.125 µl	2 µl†	4
DNA Template (10:1)	1 µl	–	‡
FWD Primer (10 µM)	0.5 µl*	–	‡
REV Primer (10 µM)	0.5 µl*	–	‡
Water	–	199.5 µl	1
	25 µl	325 µl	

Table B.1 GoTaq reaction mix, used for initial exploratory PCR.
(†) Polymerase volume rounded to ease pipetting, (‡) DNA and primers added to individual reaction tube after splitting master mix, (*) Primer at 0.2 µM in individual reaction

Denature	**Denature**	**Anneal**	**Extend**	×30	**Extend**	**Store**
`95°C`	`95°C`	`58°C`	`72°C`		`72°C`	`4°C`
`5:00`	`0:20`	`0:20`	`1:40`		`5:00`	∞

Table B.2 Thermocycler program for initial exploratory PCR

Brief Protocol

- Produce master mix as per Table B.1 consisting of water, buffer, dNTPs and polymerase. On ice.
- Split master mix into 13 PCR reaction tubes.
- Add specific reverse transcribed cDNA and corresponding primers to reaction tubes for each of the 10 `Gretel` regions (Table 5.7).
- Add reverse transcribed cDNA for "RT mix" positive control, and kit hex-mer primer pair.
- Add reverse transcribed cDNA for 16S positive control, and `F968`/`R1401` 16S primer pair [349].
- Add negative control cDNA and replace allocated primer volume with water (1 µl).
- Cycle as per Table B.2

Notebook number	Continued from page number	Page number 42

Date

T — purple RT STOCK
— blue PCR
— yellow — RT working (10:1)
green + yellow

		PCR III (QTest 1)	1x	x13
1	11	GoTaq Green buffer	5 µL	65 µL
2	251	dNTPs	2.5 µL	32.5 µL
3	31	GoTaq Polymerase	0.125µL	2µL (1.625 µL small)
4	~~85~~ 90 T	DNA Template (10:1)	1µL	
5	107	Fwd Primer (10µM)	0.5 µL (0.2µM)	6.5µL per reaction
6	123	Rev Primer (10µM)	0.5µL (0.2µM)	
7	142	Water		199.5
8	152		25µL	325µL
9	162	T/p23 QTest Appendix C; as more than ½ of PCR vol should derive from the RT product		
10	165			to 20µL should use ≤ 2µL
+11	Toby	* Epoch estimates ≈ 0.8–1 µg/µL, let try a 10:1 min		
+12	RT Mix			
−13	empty	— 23µL of mix into 13 PCR tubes (BLUE)		
		— 1µL of each 13 DNA		
		— 0.5 µ of each 10µM primer, except #12 1µL RT mix #13 1µL water.		

(accidentally added 1µL water to 12 and 13 also)

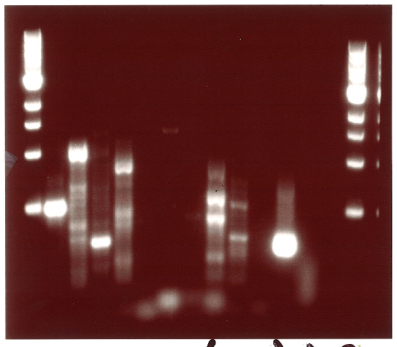
OK

Performed by	Date	Countersigned by	Date	Continued on page number

Notebook number	Continued from page number	Page number 43

Date ~~4/3~~ PCR G...E...

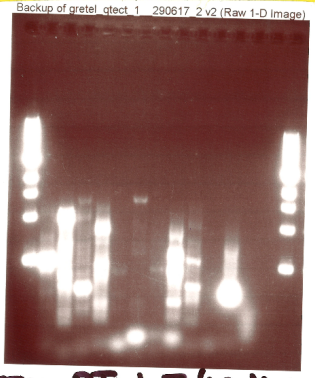
gretel_qtect_1__290617 v2 (Raw 1-D Image)



GRETEL QTest I (10:1) \2

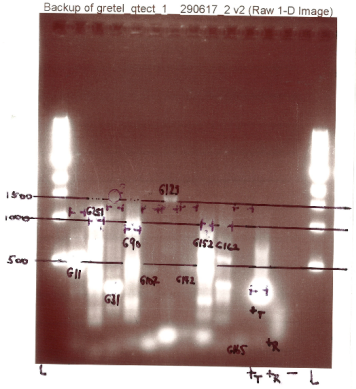
GRETEL QTest I (10:1)

Backup of gretel_qtect_1__290617_2 v2 (Raw 1-D Image)



GRETEL QTest I (10:1)\1

Backup of gretel_qtect_1__290617_2 v2 (Raw 1-D Image)



Performed by	Date	Countersigned by	Date	Continued on page number
	5/7/17			

Notebook number	Continued from page number	Page number 46

Date 6.17 PCR Gradients

4x100µL

	(25µL) x1	(100µL) x4	x4 Master Mix
PCR 100µL MM			
Gotaq Buff Grn	5µL	20µL	80µL
dNTPs	2.5µL	10µL	40µL
Gotaq Polymerase	0.125µL	0.5µL	2µL
DNA Template	1µL	4µL	—
Fwd	0.5µL	2µL	—
Rev	0.5µL	2µL	—
Water	15.5µL	246	86µL (86+160)
	G1		100

PCR 100µL	(100µL) x1	1	2	3	4
MasterMix	92µL				
Template (10:1)	4µL	1	2	4	6
Fwd	2µL	11 F	251 F	90 F	123 F
Rev	2µL	11 R	251 R	90 R	123 R
	100µL				

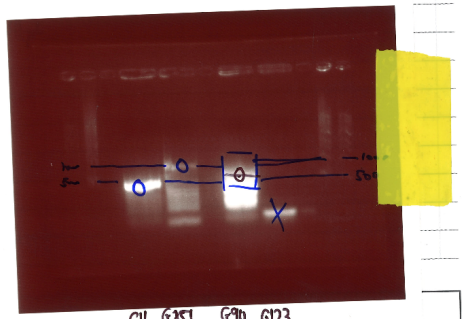
G	11	251	90	123
	1	2	3	4
	5	6	7	8
	Anneal 60°C		Anneal 55°C	

1 2 3 4

Ladder	1x	2x
H_2O	4µL	8µL
Dye	1µL	2µL
Ladder	1µL	2µL
	6µL	12

Thr ~ 1.029g
1.0469g

1 1.5904 = 0.54
2 1.8716 = 0.8926
3 1.3404 = 0.3114



Performed by	Date

Notebook number	Continued from page number	Page number 47

Date 6.17 Ext. extra

			x3 QG		
1	561.4mg ∴		561.4mg	1.6842	1684.2µL = 842x2µL
2	842.6mg ∴	1.376	334.4mg	1.0032	1003.2
		1.320	278.4mg	0.8352	835.2µL
3	311.4mg ∴		311.4mg	0.9342	934.2µL

① QG

② 10min @ 50°C

		x1 Isopropanol
③ 1	1A 1mL	280.7µL
	1B 1mL	
2A		334.4µL
2B		278.4µL
3		311.4µL

④ elute ~500?µL to 1.5mL spin column — make sure not to touch the bottom of column "the matrix"
⑤ 750µL PE & centrifuge
⑥ & again
⑦ transf to 1.5mL tube, leave open to allow ethanol to evaporate

⑧ 25µL water & spin — water must go on the matrix (x2 — 8 to make 50µL)
⑨ spin — & watch the caps don't break off (they did)

⑩ Equit. ~ 1 ~10ng/µL = 0.5µL (~5ng)
2 ~3ng/µL = 1µL (~3ng)
3 ~10ng/µL = 0.5µL (~5ng)

Sequencing Reaction ⑬	(6µL) x1	x2	(12µL) x4
Template	0.5µL	1µL	4µL
Primer Fwd (1.6pmol)	1.6µL	3.2	8µL
Rev seq kit	4µL	8	
Water	3.9µL	7.8µL	20µL

Need a 1:10 dilution of the 10µM stock = 5µL primer + 45µL water ∴ 1µM

⑫	1µL	2µL
	1.6	3.2
	4	8
		6.8

	1	2	3	4	5	6
	11 F	11 R	251 F	251 R	90 F	90 R

Performed by	Date	Countersigned by	Date	Continued on page number

Notebook number	Continued from page number	Page number 48

Date 10/7

PCR Gradient Master Mix — 25µL x1 — x10 — x5

	25µL x1	x10	x5
Buffer Green	5µL	50µL	250µL
dNTPs	2.5µL	25µL	1,25µL
Polymerase	0.125µL	1.25µL	6.25µL
Template	1µL	10µL	
Fwd	0.5µL	5µL	
Rev	0.5µL	5µL	
Water	15.375	240.375	

— made new 2mM dNTP stock
245µL MQ & 5µL 100mM stock

PCR Gradient Mix	25µL x1	(Rep) x1	Sperm x10	Neptune x5
Buffer Green	5	4	40	200
dNTPs	2.5	2	20	100
Polymerase	0.125	0.1	1	5
Template	1	0.8	8	40
Fwd	0.5	0.4	4	20
Rev	0.5	0.4	4	20
Water	15.375	12.3	123	615
		20	200 ✓	1000 ✓

1 | 2 | 3 | 4 | 5
x1: 31 90 123 152 251

1 — 184µL neptune
2 — 3 4 6 8 2
3 — 40
4 — 20
5 — 40
6 — 20
7 — 20

purple → yellow — 10:1 DNA v2

- Sperm
- 20µL into 10x buffer & PCR

L <10> L <10> L

$L \cdot \frac{1}{10} \times L \times \frac{1}{10} L$

$L \cdot \frac{1}{10} \times \frac{1}{10} L$

Performed by	Date	Countersigned by	Date	Continued on page number

Notebook number	Continued from page number 48	Page number 49

Date

B (1) — LL — <1> <2> — XL — <3> <4>

<10> X L X <10> X L X <10> X L X <10> X L X <10> X L

Comb 1 (row 2)

(5µL of all)

① <10> L <10> L
② <10> L <10> L
③ <10> L <10> L
④ <10> L <10> L

120V

G (11) TANK — L — <5>

X X X L <10> L

gretel-gradient1_2017_07_10 (Raw 1-D Image) — 7_10_num5 v2 (Raw 1-D Image)

1 G31 | 2 G90 | 3 G123 | 4 G152 | t1 | G251

52°C

62°

Sam

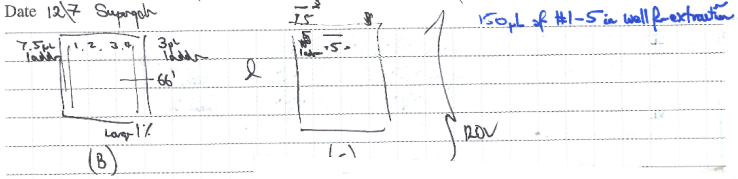
B.2 Gel Extraction

The following pages are scanned from my laboratory notebook, showing the record keeping for gel extractions. Extractions were performed with the *Qiagen QIAquick Gel Extraction Kit* according to manufacturer's instruction with the following modifications:

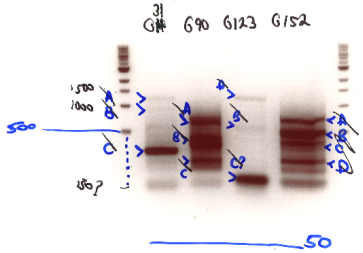
- After isopropanol spinning step, we take the protocol's suggestion of adding 500 µl Buffer QG and centrifuging the columns.
- After the addition of Buffer PE during the washing step, we stand the tube for 5 minutes.
- Buffer EB is replaced with water at the wash step to avoid EDTA affecting nanopore sequencing.
- The 50 µl wash (with water instead of Buffer PE) is conducted in two 25 µl wash and spin steps in an attempt to improve yield. The tube stands for 2.5 minutes after addition of water.

Notebook number	Continued from page number	Page number
9	48	50

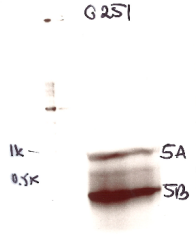
Date 12/7 Superguide



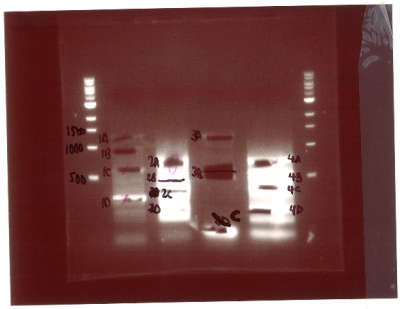
samples_1-4_white (Raw 1-D Image)



sample5 (Raw 1-D Image)



Administrator 2017-07-12 15hr 17min (Raw 1-D Image)



Performed by	Date	Countersigned by	Date	Continued on page number
[redacted]	12/7			

Notebook number	Continued from page number	Page number
	50	51

Date 12/7 Gel Extraction (-1.1069) x1

		ng		X3 QG	Yellow?	x1 Isop.	spin	#	Tube weight
1	1A	1.2804	173.5mg	520.5			2		
2	1B	3524	245.5	736.5			2		
3	1C	3332	226.3	678.9			2		
4	1D	2870	180.1mg	540.3			2		
5	2A	2768	169.9	509.7			1		
6	2B	2156	108.7	326.1			1		
7	2C	3325	225.6	676.8			2		
8	2D	1923	85.4	256.2			1		
9	3A	3237	216.8	650.4			2		
10	3B	4501	343.2	1,029.6			3		
11	3C	2733	165.8	497.4			2		
12	4A	3853	278.4	835.2			2		
13	4B	2225	115.6	346.8			1		
14	4C	2700	163.1	489.3			1		
15	4D	2131	106.2	318.6			1		
16	5A	3402	233.3	699.9			2		
17	5B	3888	281.9	845.7			2		

50°C ~10m — 13 Krpm · 1min

1A	2.9875	1.375	0.575	1.975	~2	5A	4.575	5.275	~5
1B	1.875	3.775		~2		4A	3.475	9.375	~3.4
1C	1.675	2.775		~2		4B	4.975	4.875	~4.9
1D	6.875	7.875		~7		4C	2.875	2.775	~2.75
2A	10.575	16.375	3.075	0.675	3.775 ~8.5	4D	1.775	2.575	~2
2B	4.675	5.875		~4.8		5B	11.075	6.775	11.975 17.575 ~10?
2C	10.775	6.775	5.775	6.175	~6				
2D	3.475	3.575		~3.5					
3A	4.075	1.775	0.175	7.875	0.775 1.975				
3B	2.475	2.775		~2.5					
3C	1.175	0.075	1.775	1.675	~1.6				

NGS Sequencing Mix		6µL x1	12µL x2
Template			
Primer (1.6 pmol/µL)		1.6µL	3.2µL
IGER Kit		4µL	8µL
Water			

B.3 Sanger Sequencing

The following pages are scanned from my laboratory notebook, showing the reaction mix and record keeping for samples submitted for Sanger sequencing.

Template Size	Required Mass
100 – 200 bp	0.5 ng – 1.5 ng
200 – 500 bp	1.5 ng – 5 ng
500 – 1000 bp	2.5 ng – 10 ng
1000 – 2000 bp	20 ng – 50 ng

Table B.3 Facility template quantity guidelines for use of ABI 3730 DNA analyser

Sanger `2017-09.S1`	1×
Terminator Ready Reaction Mix	4 µl
Template	see Table B.3
Primer	1.6 pmol
Water	*q.s.*
	10 µl

Table B.4 Facility guidelines for preparation of Sanger sequencing samples

Notebook number	Continued from page number 51	Page number 52

Date 13/7 Sequencing Mix

	Sequencing Mix / template	~conc	needs	have	add	(10:1) 6xL† Primer 3.2	x2 Kit 8	Σ	Water	Σ
1	1A	~1250 ...	5-10ng	~2	5µL	"			3.8	
2	1B	~900 ..	2.5-10	~2	2µL 5	"	G31†		"	
3	1C	~600 .	2.5-10	~2	5	"			"	
4	1D	2250	≤5	~7	10 5	"			7.3	
5	2A	~750 .	2.5-10	~3.1	3	"			5.8	
6	2B	~500 .	≤5	~2.5	2	"			6.8	
7	2C	~500 .	≤5	~6	1.5	"	G90		7.3	
8	2D	~500 .	≤5	~3.5	2.5	"			6.3	
9	3A	~1100 ...	5-20	~2	5	"			3.8	
10	3B	~500 .	2.5-10	~2.5	4	"	G123		4.8	
11	3C	~500 .	≤3	~1.6	3.5	"			5.3	
12	4A	~600 .	2.5-10	~3.4	3	"			5.8	
13	4B	~500 .	2.5-10	~4.9	2	"	G152†		6.8	
14	4C	~500 .	≤5	~2.75	3.5	"			5.3	
15	4D	~500 .	≤5	~2	5	"			3.8	
16	5A	~1000 ..	2.5-10	~5	2	"	G251		6.8	
17	5B	~500 .	≤5	~10	0.5µL	"			8.3	

† needs 10:1 primer mixed (might need more?)
F will need to be made up

green tube

Performed by	Date	Countersigned by	Date	Continued on page number

Notebook number	Continued from page number 52	Page number 53	52B

Date

W_1	W_2	T_1	T_2	Fwd P_1	Rev P_2	Cassette	dna conc.	Blunt	hpc?	
1.8	1	5	5	1	2	? ?	2	\ \	x	FR
2.8	3	5	5	3	4	? ?	2	\ \	x	FR
3.8	5	5	5	5	6	/ /	2	x x	x	
7.3	7	1.5	1.5	7	8	? /	7	\ x	x	FR
5.8	9	3	3	9	10	// //	3.5			
6.8	11	2	2	11	12	/ /	9.5	x		
7.3	13	1.5	1.5	13	14	// /	6			
6.3	15	2.5	2.5	15	16	/ /	2.5	• X	x	
3.8	17	5	5	17	18	// ?	2		☺	RF
4.8	19	4	4	19	20	? ?	2.8	\ \	x	
5.3	21	2.5	2.5	21	22	? ?	1.6	\ \	x	
5.8	23	3	3	23	24	// /	3.4			
6.8	25	2	2	25	26	// //	4.9	ribosome	?	carry over?
5.3	27	2.5	2.5	27	28	/ /	2.75	x	x	
3.8	29	5	5	29	30	/ /	2	x	x	
6.8	31	2	2	31	32	// //	5		☺	
5.3	33	0.5	0.5	33	34	// //	10			

31 96 123 152 251

done ☺!

Performed by	Date	Countersigned by	Date	Continued on page number

B.4 Amplification with High Fidelity Polymerase

The following pages are scanned from my laboratory notebook, showing my attempts towards reliable amplification of the `Gretel` candidates with High-Fidelity polymerase. Initial poor banding was caused by an annealing temperature that was too low.

PCR `2017-09-14.HF2`	1×	13×	Order
5× Phusion HF Buffer	10 µl	130 µl	2
dNTPs (10 mM)	5 µl	65 µl	3
DMSO	1.5 µl	19.5 µl	4
Phusion Polymerase	0.5 µl	6.5 µl	5
DNA Template (10:1)	1 µl	–	‡
FWD Primer (10 µM)	2.5 µl*	–	‡
REV Primer (10 µM)	2.5 µl*	–	‡
Water	27 µl	351 µl	1
	50 µl	650 µl	

Table B.5 Phusion reaction mix, used for high-fidelity PCR.
(‡) DNA and primers added to individual reaction tube after splitting master mix, (*) Primer at 0.5 µM in individual reaction

Denature	**Denature**	**Anneal**	**Extend**	×30	**Extend**	**Store**
`98°C`	`98°C`	`65°C`	`72°C`		`72°C`	`4°C`
`0:30`	`0:10`	`0:20`	`1:00`		`5:00`	∞

Table B.6 Thermocycler program for high-fidelity PCR

Brief Protocol

- Produce master mix on ice, as per Table B.5.
- Split master mix into 13 PCR reaction tubes (44 µl per tube).
- Add specific reverse transcribed cDNA and corresponding primers to reaction tubes for each of the 10 `Gretel` regions (Table 5.7).
- Add reverse transcribed cDNA for "RT mix" positive control, and kit hex-mer primer pair.
- Add reverse transcribed cDNA for 16S positive control, and `F968`/`R1401` 16S primer pair [349].
- Add negative control cDNA and replace allocated primer volume with water (5 µl).
- Cycle as per Table B.6

Notebook number
Page number 60
Date 8.9 Amplification
Components
DMSO
Water
Polymerase
Template (gBlock ID1)
Rev 10µM
Fwd 10µM
10µM mix
Buffer
Notebook number
Continued from page number
Page number 61
Phusion® High-Fidelity DNA Polymerase
M0530S
100 units
2,000 U/ml
Lot: 0051704
Exp: 4/19
Store at -20°C
Quick-Load® Purple 100 bp DNA Ladder
N0551S
125 gel lanes
50 µg/ml
Lot: 0061704
Exp: 4/19
Store at 4°C
Protocol for a Routine PCR
Storage Notes: This product is stable for at least 6 months at 25°C. For long term storage, store at 4°C or -20°C. If stored at -20°C, mix well after thawing.
BioLabs
NEW ENGLAND
Performed by
Date
Countersigned by
Date
Continued on page number
hifi1 (Raw 1-D Image)
hifi white (Raw 1-D Image)

Notebook number
Continued from page number
Page number 62
Date
Performed by
Date
45ul_1234689_white (Raw 1-D Image)
Notebook number
Continued from page number 62
Page number 63
Date
16-Sep-2017 11:54:33 Low=0 High=614 Gamma=3.0 Exposure = 1.024 secs
11:55:31 Low=0 High=2861 Gamma=2.0 Exposure = 1.024 secs

Notebook number | Continued from page number | Page number 66

Date Sun 12/9/17

45 → 60 s
- extension
- 58°C → 65°C

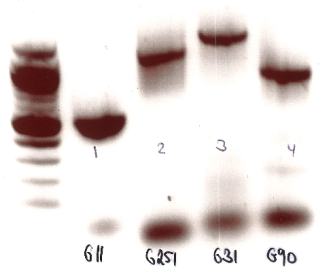
Hifi (25 µL)	1x	5x
5x Buffer	5	25
dNTP	0.5	2.5
Polymerase	0.25	1.25
DMSO	0.75	3.75
(a) Σ	6.5	
Water (c) (55 − (a+b))	15	75
MASTER MIX (mix) (a+c)	21.5	
FWD	1.5	
REV	1.5	
T		
(b) Σ	3.5	

	1	2	3	4	5
	∅	∅	∅	∅	∅
	✓	✓	✓	✓	–
	✓	✓	✓	✓	–
	✓	✓	✓	✓	–

98	98		72	72	
30	10	65 (−0.5°C/cycle)	60	5	4 ∞

30x

2017-09-17_pcr1234_success_white (Raw 1-D Image)



Pl... | Continued on page number

Notebook number | Continued from page number | Page number 67

Date Mon 18/9

Hifi (50 µL)	1x	8x	8x2
5x Buffer	10	80	160
dNTP	5	40	80
Polymerase	0.5	4	8
DMSO	1.5	12	24
(a) Σ	17		456
Water (c) (50 − (a+b))	28.5	228	
MASTER MIX (mix) (a+c)	45.5		
FWD	1.5		
REV	1.5		
T	1.5		
(b) Σ	4.5	36	

- Span DMSO to remove fluor
- add 20µL from purification to 1-4
- avoid freeze
- ~~use EB?~~ NO
- elute in 100 nuc dNTPs

Σ400 ✓ Σ800 ✓

98	98		72	72	
30	10	65 (−0.5°C/cycle)	60	5	4 ∞

30x

$V_1C_1 = V_2C_2$

$Vol_{stock} \cdot Conc_{stock} = Vol_{working} \cdot Conc_{working}$

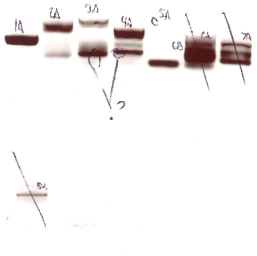
$x \cdot 100\,mM = 250\,\mu L \times 2mM$

$x = \frac{500}{100}$

$x = 5\,\mu L$ in 245 µL

Gel
- 1.5µL loading
- 7.5µL gel (1% gel)
- 120V

2017-09-18_bigpcrhifi (Raw 1-D Image)



Performed by | Date

Notebook number	Continued from page number	Page number 68

Date 18/9

1.1053
1.1065
1.1054
μ 1.1067

#	g	x1 base	x3 QC	Xbuf	Isop	Spin	500 QC	spin	750 FE	stand	spin	stand	Supt stand (4m) spin			
1·2	1.3853	278.6	835.8											12.6	11.77	(2)
2·2	1.3432	236.5	709.5											6.77	6.87	(2)
3·2	1.3471	240.4	721.2											2–3?		(2)
4·2	1.3103	203.6	610.8											~~5–12?~~		~~(2)~~
5·2	1.3613	254.6	763.8											2.2	2.6	(2)

1/9/9

#	g	x1 base	x3 QC	Mast mix	Xbuf	Isop	x1	Spin	500 QC	spin	750 FE	stand	spin	stand	Supt stand 4m spin ①	②	repeats ①	④	⑤
3·2	1.3387	227.0mg	681.0									3		1			6.05	7.35	6.65
5·2	1.3186	211.9mg	635.7									3		1			12.15	13.85	13.25

2/9/9 Qubit

Reagent 5.2μl
Qubit Buffer 1034.8
working solution
5 tube samples
Read contents

#	g	x1 base	x3 QC	y	x1 base	spin	500 QC	spin	750 FE	stand	stand	stand	20 DNA	20 DNA	20 DNA	Dilute spin
1·3	1.4053	298.6	895.8													
2·3	1.2712	164.5	493.5													
3·3	1.2623	155.6	466.8													
4·9	1.3946	287.9	863.7													
5·3	1.3757	269.0	807.0													

Qubit Stored

Performed by	Date	Countersigned by	Date	Continued on page number

Notebook number	Continued from page number	Page number 69

Date 18/9 Spin

	1x	5x	
Hyb (50μL)	50	50	– spin DMSO
Buffer	10	50	~~– use 10 NTP used~~ new dNTP
dNTP	5	25	~~not new 10?~~ ~~used yellow (6)~~
Poly	0.5	2.5	– using new H_2O (11.2mg)
DMSO	1.5	7.5	
Σ	17		
WATER	285	142.5	
MN	45.5		
fwd	1.5		
rev	1.5		
t	1.5		
Σ	45.5		

3 5
1 2 3 4 –
1 2 3 4 –
1 2 3 4 5

3.2.2
5.2.2

285 + 17

not (6), new (5)

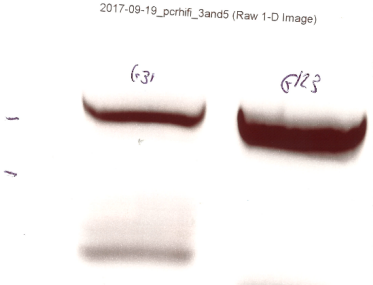
250 √6

green | ~~yellow~~ | purple

DNA 10:1 v2

– 100V
– 2.5% ladder
– 2.5% agarose

gel good extract



Performed by	Date	Countersigned by	Date	Continued on page number

Notebook number	Continued from page number	Page number 72

Date 2/9

#	approx bp	Qubit ng/μL	nanodrop	~ng			
1	500	9.4 9.48	~~0.2~~	~9.4	=	31.915	0.91 pmol
2	1000	2.32 2.20	~~0.5~~	~2.2	=	136.36	0.45
3	1500	~4	~~0.5~~	~4	=	75.00	0.3
4	1000	3.52 3.53	~~0.2~~	~3.5	=	85.714	0.45
5	1000	10.6 10.6	~~0.5~~	~10	=	30.00	0.45
						358.979	

(300ng) 0.3μg need ng/μL μL

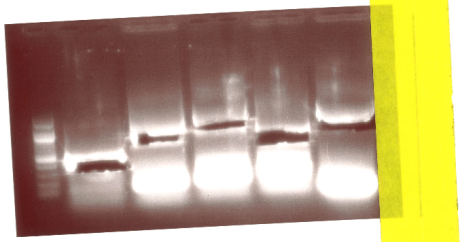
#) This seems too short - Product is supposed to be ~900!

Hyb (50μL)	1x	10x	11x
5x Buffer	10	100	110
dNTPs	5	50	55
Polymerase	0.5	5	5.5
DMSO	1.5	15	16.5
	Σ 17		
Water	28.5	~~285~~	313.5

Next Nunc DNA

		1	2	3	4	5	6	7	8	9	10	11
MM	45.5	○	○	○	○	○	○	○	○	○	○	○
Fwd	1.5	~~11~~	~~11~~	251	251	31	31	90	90	113	113	—
Rev	1.5	11	11	251	251	31	31	90	90	113	113	—
T	1.5	1A	1A	2A	2A	3A	3A	4A	4A	5A	5A	—

2017-09-20_repeat_cut (Raw 1-D Image)



Performed by	Date	(

Notebook number	Continued from page number	Page number 73

Date

Hyb (25μL)	1x	5x	(5)	(25)	(26)
5x Buffer	5	25		5x	+1
dNTPs	2.5 (0.5)	~~12.5~~		12.5	130
Polymerase	0.25	1.25		62.5	65
DMSO	0.75	3.75		6.25	6.5
	(a) Σ 8.5			18.75	19.5
Water (c)	25 (Conc)	13			338
Master Mix (arc)	21.5				

21.5 ○ ○ ○ ○ ○ ○ ○ ○ ○ ○

Master Mix 1 2 3 4 5

Master Mix 21.5

= 37 cycles

- EB

		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Master Mix	21.5															
Fwd	1.25															
Rev	1.25															
T	1															

		16	17	18	19	20	21	22	23	24	25	26
Master Mix	21.5											
Fwd	1.25											—
Rev	1.25											—
T	1											—

- should have mixed master mix during settled period...

	1	2	3	4	5	6	7	8
A								
B								
C								
D								

E650μL

p68

p68 for extract

Performed by	Date	Countersigned by	Date	Continued on page number

B.5 Oxford Nanopore Sequencing

B.5.1 Amplicon Pooling

The following pages are scanned copies of my laboratory notebook, demonstrating the Qubit Fluorometric assessment (left) of the `Gretel` amplicons extracted from the gel, and the calculation of molarity (right) for the sequencing experiment.

Notebook number	Continued from page number	Page number 74

Date 21/9

Qubit

	r1	r2		
1 v2	9.40	9.48		~9.4
2 v2	2.23	2.20		~2.2
3 v2	5.20	4.08	(5.6)(4.28)	~4
4 v2	3.52	3.53		~3.5
5 v2	10.6	10.6		~10.6
1 v3	1.58 EQ4	(381)	15.4	~15.5
2 v3	1.92 EQ3	(97.1)	1.88	~1.9
3 v3	2.01 EQ3	(50.5)	2.02	~2
4 v3	5.76 EQ3	(143)	5.72	~5.7
5 v3	9.52 EQ3	(132)	9.32	~9.5

Qubit (ng/μL) - 5μL S2S.
1- 9.40, 9.48
2- 2.23, 2.20
3- 5.20, 4.08, 5.6, 4.28
4- 3.52, 3.53
5- 10.6, 10.6
3+10 min = 3.72, 4.12, 3.66, 5.76, 3.56, 3.56, 3.56

SAM'S SAMPLES!

Qubit
~~Reagent~~ ~~5~~
Buffer ~~199~~ 995 } working

Sample 5μL
working 195μL

21/9 Gretel Soya MkI. (see p72.)

1.5μg \5 = 300ng
500bp @ 300ng = 0.91pmol
1000bp @ = 0.45pmol (x2.02)
1500bp @ = 0.36pmol (x2.52)
850bp @ = 0.30pmol (x3.03)

probably close enough to just ratio:.

	#1	#2	#3	#4	#5	
	500 :	1250 :	1500 :	1000 :	1500	
	1 :	2.5 :	3 :	2 :	3	Σ11.5
	130.4	326	391.2	260.8	391.2	Σ[1499.6ng]

assuming/acting average content

Σ 1499.6ng

Performed by	Date	Countersigned by	Date	Continued on page number

Notebook number	Continued from page number	Page number 75

Date 22/9

# v	~ng/μL	TARGET	300ng μL required	÷2ng	μL	
1-2	9.4	130.4ng	13.872	6.936	6.936	6.94
2-2	2.2	326 ng	148.18	(*) 89.09 (cap at 85)	(*)85	85.0
3-2	4.0	391.2ng	97.8	48.9 +3.9(†)	52.8	52.8
4-2	3.5	260.8ng	74.51	37.255	37.255	37.3
5-2	10.6	391.2ng	36.906	18.453	18.453	18.45
1-3	15.5	130.4ng	8.443	4.2065	4.206	4.2
2-3	2.9	326 ng	112.414	56.207 +3.102	59.309	59.31
3-3	2.0	391.2ng	195.6	(*) 97.8 (only have 95μL)	90	90
4-3	5.7	260.8ng	45.75	22.875	22.875	22.88
5-3	9.3	391.2ng	42.06	21.03	21.03	21.0

(†) 3.3 at 90μL, needs 7.8μL × 2ng = 15.6ng
3.2 $\frac{15.6ng}{4.0ng/\mu L}$ = 3.9μL extra

(θ) 2.2 at 85μL needs 4.09 × 2.2ng = 8.998ng
2.3 $\frac{8.998ng}{2.9}$ = 3.102μL extra

Σ 397.8?

	μL	×10%	Soya μL	Added to pool
1-2	6.936	7.6296	= 7.63	− 70% EtOH
2-2	85	93.5	= 93.5	− 0.6 SPRI bead concentration
3-2	52.8	58.08 +5 = 63.08	= 63.1	
4-2	37.255	40.9805	= 41.0	
5-2	18.453	20.2983	= 20.3	
1-3	4.206	4.6266	= 4.63	
2-3	59.309	65.2399	= 65.24	
3-3	90	~~99~~ (x)	= 90	
4-3	22.875	25.1625	= 25.2	
5-3	21.03	23.133	= 23.2	
			Σ 433.8	~433.8μL (MKI)

Performed by	Date	Countersigned by	Date	Continued on page number

B.5.2 SQK-LSK108 Sequencing Preparation

The following included pages are a copy of the Oxford Nanopore SQK-LSK108 log sheet that I used during the preparation and long-read sequencing of the "Gretel Soup" sample.

msn 22/9/17

1D Genomic DNA sequencing for the MinION™ device using SQK-LSK108

Oxford NANOPORE Technologies

Flow Cell Number: ..... DNA Samples: .....

Before start checklist

- [x] Ligation 1D Sequencing Kit SQK-LSK108 (AMX 1D in freezer until needed)
- [x] NEB Blunt/TA Ligase Master Mix
- [ ] Notebook sleeptimer/update off
- [x] AMPure beads
- [x] Heat block for 1.5ml tubes at 37°C
- [ ] User adapt loading for (FLO-MIN104)
- [x] Microfuge
- [x] Thermal cycler at 20°C and 65°C
- [x] NEBNext Ultra II End-repair/dA-tailing module
- [x] Vortexer and Hula mixer
- [x] Magnetic rack for bead separation
- [ ] Pre-chilled freezer pack for 0.2ml tubes at -20°C
- [x] Pipettes P2, P20, P100/200, P1000
- [ ] Nuclease-free water (NFW) (not supplied)
- [ ] DNA LoBind Eppendorf tubes 1.5ml
- [x] Pipette tips P2, P20, P100/200, P1000
- [x] 10mM Tris-HCl pH8.5 (if used)
- [x] 0.2ml thin-walled PCR tubes
- [ ] Freshly prepared 70% EtOH
- [ ] Platform QC completed on flow cell
- [x] 0.2ml thin-walled PCR tubes Latest versions of MinKNOW and Desktop Agent (check for updates)
- [x] MinION SpotON FLO-MIN106 or FLO-MIN107

Additions to checklist for this protocol:

Covaris g-TUBE (optional), NEBNext FFPE Repair Mix

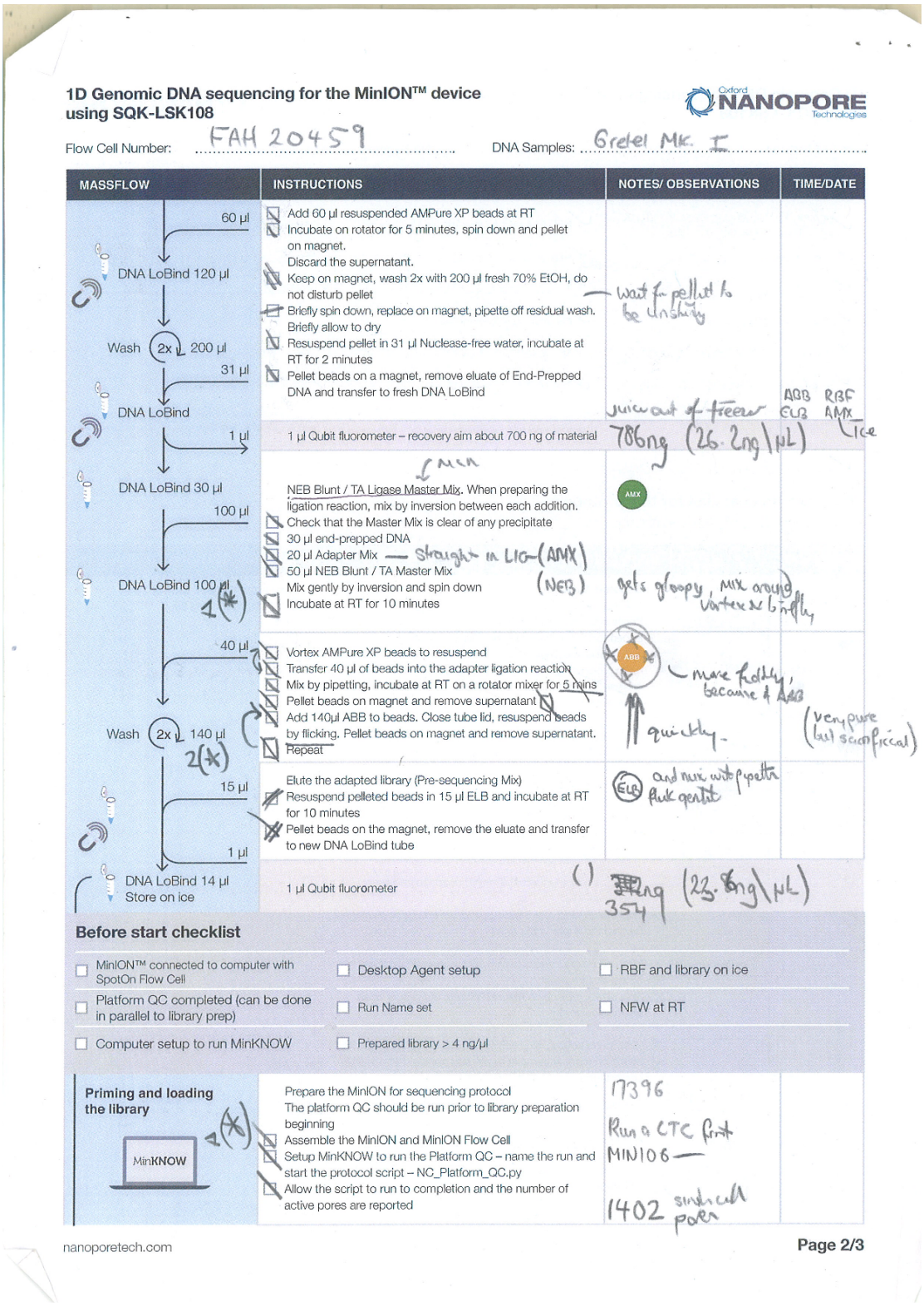
MASSFLOW	INSTRUCTIONS	NOTES/ OBSERVATIONS	TIME/DATE
DNA LoBind 46 µl 1 µl Covaris g-TUBE 45 µl	**Genomic DNA** 1-1.5 µg Genomic DNA in 46 µl Shear in a Covaris g-TUBE Eppendorf 5424; 6000rpm for 8 kb fragments. 2 x 1 minute (invert tube to collect) Keep processing time under 15 minutes Analyse 1 µl for fragment size, quantity and quality (Agilent Bioanalyser)	This step is optional and can be omitted if long fragments are required.	
DNA LoBind 45 µl 17 µl	DNA repair (optional) Add from NEBNext FFPE RepairMix (NEB M6630) ☐ 8.5 µl NFW ☐ 6.5 µl FFPE Repair Buffer ☐ 2 µl FFPE Repair Mix		
DNA LoBind 62 µl 62 µl	Mix by inversion + spin down Incubate for 15 mins at 20 °C		
DNA LoBind 124 µl	☐ ~~Add 62 µl resuspended AMPureXP beads to end-prep reaction at RT~~ ☒ Incubate on rotator for 5 mins, spin down and pellet on magnet Discard the supernatant at RT		
Wash 2x 200 µl	Keep on magnet wash 2x with 200 µl fresh 70% EtOH, ☒ do not disturb pellet ☒ Spin down, replace on magnet, pipette of residual wash Briefly allow to dry		
46 µl DNA LoBind 1 µl	☒ Resuspend pellet in 46 µl NFW, incubate for 2 mins at RT Pellet ☒ beads on magnet, remove eluate and transfer to fresh DNA LoBind tube	on rack	lab's put aside Tube
	1 µl Qubit fluorometer-recovery ≥ 1000 ng of material	33.8ng/µL 152ng	(14.8)µL
DNA LoBind 45 µl 15 µl DNA LoBind 60 µl	Add from NEBNext Ultra II End-Repair / dA-tailing Module ☒ 7 µl Ultra II End-Prep buffer ☒ 3 µl Ultra II End-Prep enzyme mix ☒ 5 µl Nuclease-free water Mix by inversion + spin down Incubate for at 20 °C for 5 minutes and 65 °C for 5 minutes	flick tube add to PCR tube (to the incubator)	

nanoporetech.com

Page 1/3

room temp

Use 5-storm incubate mode



1D Genomic DNA sequencing for the MinION™ device using SQK-LSK108

Oxford NANOPORE Technologies

Flow Cell Number: FAH 20459 DNA Samples: Gretel MK. I

MASSFLOW	INSTRUCTIONS	NOTES/ OBSERVATIONS	TIME/DATE
60 µl → DNA LoBind 120 µl	☒ Add 60 µl resuspended AMPure XP beads at RT ☒ Incubate on rotator for 5 minutes, spin down and pellet on magnet. Discard the supernatant. ☒ Keep on magnet, wash 2x with 200 µl fresh 70% EtOH, do not disturb pellet ☒ Briefly spin down, replace on magnet, pipette off residual wash. Briefly allow to dry ☒ Resuspend pellet in 31 µl Nuclease-free water, incubate at RT for 2 minutes ☒ Pellet beads on a magnet, remove eluate of End-Prepped DNA and transfer to fresh DNA LoBind	wait for pellet to be clear	
Wash (2x) 200 µl 31 µl → DNA LoBind		juice out of freezer	ABB RBF ELB AMX Ice
1 µl →	1 µl Qubit fluorometer – recovery aim about 700 ng of material	786ng (26.2ng/µL)	
DNA LoBind 30 µl 100 µl → DNA LoBind 100 µl	NEB Blunt / TA Ligase Master Mix. When preparing the ligation reaction, mix by inversion between each addition. ☒ Check that the Master Mix is clear of any precipitate ☒ 30 µl end-prepped DNA ☒ 20 µl Adapter Mix ☒ 50 µl NEB Blunt / TA Master Mix Mix gently by inversion and spin down ☒ Incubate at RT for 10 minutes	MIX AMX Straight in LIG (AMX) (NEB) gets gloopy, mix around vortex & briefly	
40 µl →	☒ Vortex AMPure XP beads to resuspend ☒ Transfer 40 µl of beads into the adapter ligation reaction ☒ Mix by pipetting, incubate at RT on a rotator mixer for 5 mins ☒ Pellet beads on magnet and remove supernatant ☒ Add 140µl ABB to beads. Close tube lid, resuspend beads by flicking. Pellet beads on magnet and remove supernatant. ☒ Repeat	ABB more fiddly, because of ABB quickly.	(very pure but scarpical)
Wash (2x) 140 µl 15 µl →	☒ Elute the adapted library (Pre-sequencing Mix) ☒ Resuspend pelleted beads in 15 µl ELB and incubate at RT for 10 minutes ☒ Pellet beads on the magnet, remove the eluate and transfer to new DNA LoBind tube	ELB and mix with pipette flick gently	
1 µl → DNA LoBind 14 µl Store on ice	1 µl Qubit fluorometer	( ) 354ng (23.6ng/µL)	

Before start checklist

☐ MinION™ connected to computer with SpotOn Flow Cell	☐ Desktop Agent setup	☐ RBF and library on ice
☐ Platform QC completed (can be done in parallel to library prep)	☐ Run Name set	☐ NFW at RT
☐ Computer setup to run MinKNOW	☐ Prepared library > 4 ng/µl	

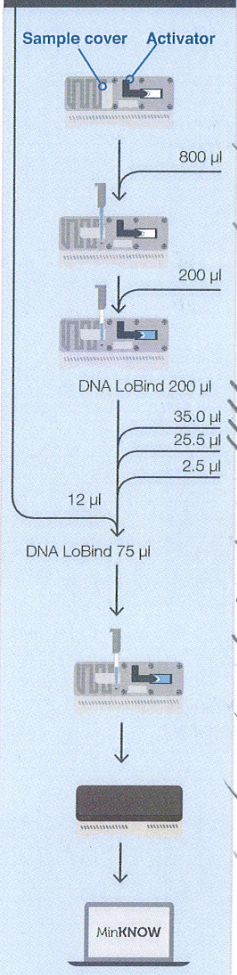
Priming and loading the library MinKNOW	Prepare the MinION for sequencing protocol The platform QC should be run prior to library preparation beginning ☒ Assemble the MinION and MinION Flow Cell ☒ Setup MinKNOW to run the Platform QC – name the run and start the protocol script – NC_Platform_QC.py ☒ Allow the script to run to completion and the number of active pores are reported	17396 Run a CTC first MIN106 1402 sinde cell pore	

nanoporetech.com Page 2/3

1D Genomic DNA sequencing for the MinION™ device using SQK-LSK108

Oxford NANOPORE Technologies

Flow Cell Number: ..... DNA Samples: .....

MASSFLOW	INSTRUCTIONS	NOTES/ OBSERVATIONS	TIME/DATE
	Prime the flow cell ready for the library to be loaded when library preparation is complete Prepare priming buffer ☑ 480 µl RBF ☑ 520 µl Nuclease-free water (RT) Prime the flow cell ☑ Open the sample port. Draw back a few µls of buffer to make sure there is continuous buffer flow from the sample port across the sensor array. ☑ Load 800µl of the priming buffer. Wait 5 minutes ☑ Gently lift the activator to make the SpotON port accessible ☑ Load 200µl of the priming buffer through the sample port Prepare the library for loading ☑ 35.0 µl RBF kept at RT ☑ 25.5 µl LLB kept at RT (helps hold down DNA) ☑ 12.0 µl Adapted and tethered library ☑ 2.5 µl Nuclease-free water Mix gently by pipetting Loading the prepared library ☑ Add 75µl of sample to the flow cell via the SpotON port in a dropwise fashion. Ensure each drop flows into the port beforeadding the next. ☑ Gently replace the activator, making sure the bung enters the SpotON port ☑ Close the sample port cover and replace the MinION li Starting the sequencing script in MinKNOW and the workflow in the Desktop Agent ☑ Return to MinKNOW, name the run, select the NC_48Hr_Sequencing_Run_FLO_MIN106_SQK-LSK108 _plus_Basecaller.py for live basecalling) using the start in the MinKNOW dialogue box ☑ MinKNOW will report the number of pores available for sequencing before data collection begins. These may differ from those reported in the Platform QC. ☑ Allow the protocol to proceed until MinKNOW reports Finished Successfully. Use the Stop in the Control Panel to finish the protocol. ☐ Quit the Desktop Agent, close down MinKNOW and disconnect the MinION After sequencing If using Albacore for local basecalling please refer to the instructions in Albacore basecalling software	(2) important to pipette accurately RBF well mixed (shake & pipette) RBF just turn pipette tip well mixed, dial down leave some in tube can be longer but not shorter RBF LLB flick mix advantage to not starting immediately	

After sequencing checklist

- ☐ Store washed flow cell at 4°C or complete the returns form in the Nanopore Community
- ☐ Return reagents to the freezer
- ☐ Navigate to **www.epi2me.nanoporetech.com** to review the full sequencing report
- ☐ Store MinION at RT

Appendix C

Additional Circos Plots

The following section includes additional `Circos` plots to illustrate the identity of `Gretel` recovered haplotypes to sequenced Nanopore reads for three additional genes: G31, G90 and G251.

C.0.1 G31

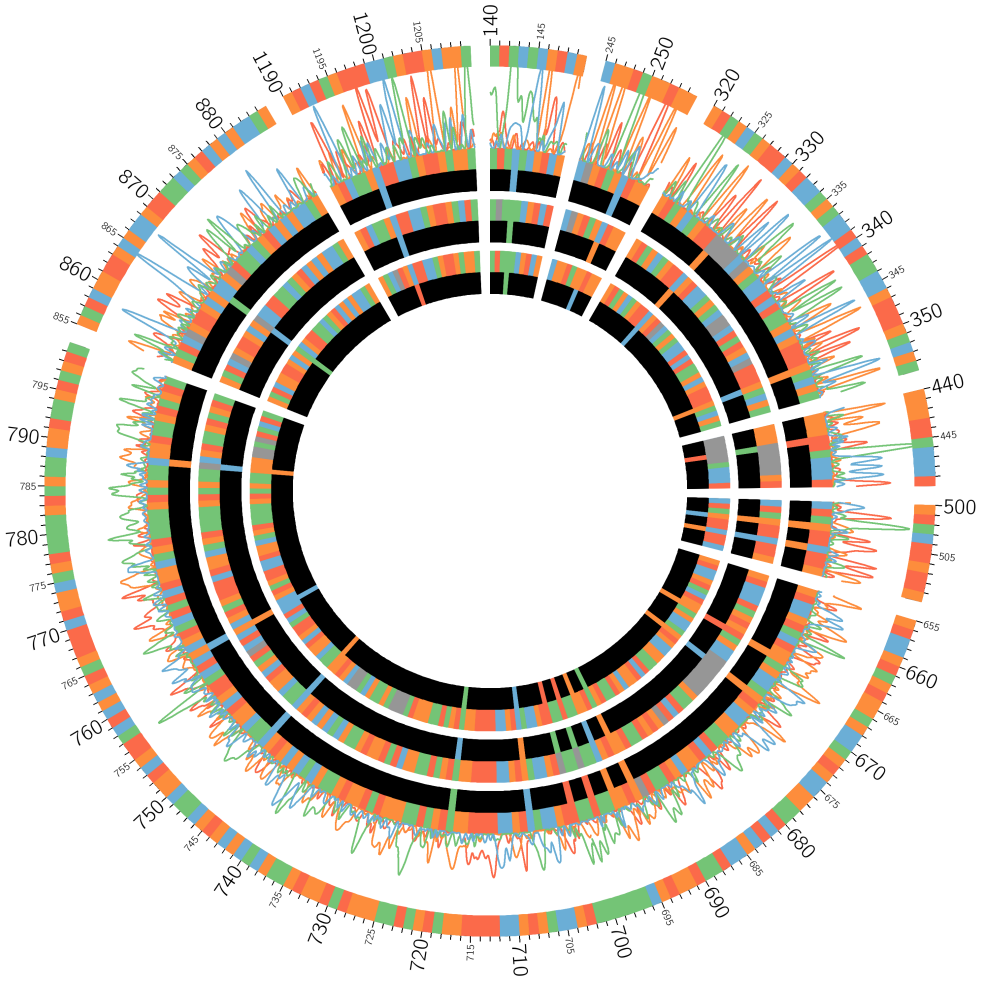


Figure C.1 `Circos` plot demonstrating `Gretel` haplotype recovery results for G31. Sites where a haplotype appears incorrect according to its corresponding Nanopore read (*e.g.* 350, 1196) still have evidence in the Sanger chromatogram, indicating a possible lack of sequencing depth to sequence the haplotypes from the amplicons or potential sequencing error.

C.0.2 G90

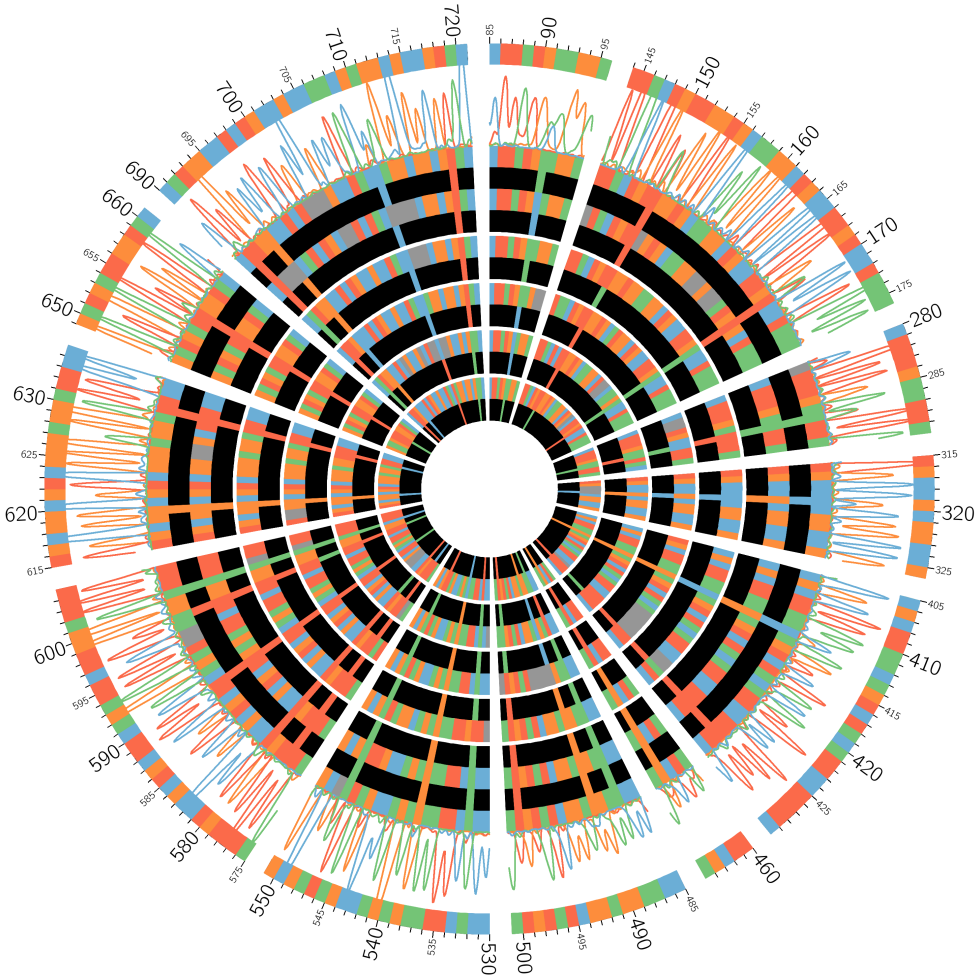


Figure C.2 `Circos` plot demonstrating `Gretel` haplotype recovery results for G90. Sites where a haplotype appears incorrect according to its corresponding Nanopore read (*e.g.* 151, 580) are still supported by the Sanger chromatogram, indicating a possible lack of sequencing depth to sequence the haplotypes from the amplicons, or sequencing error.

C.0.3 G251

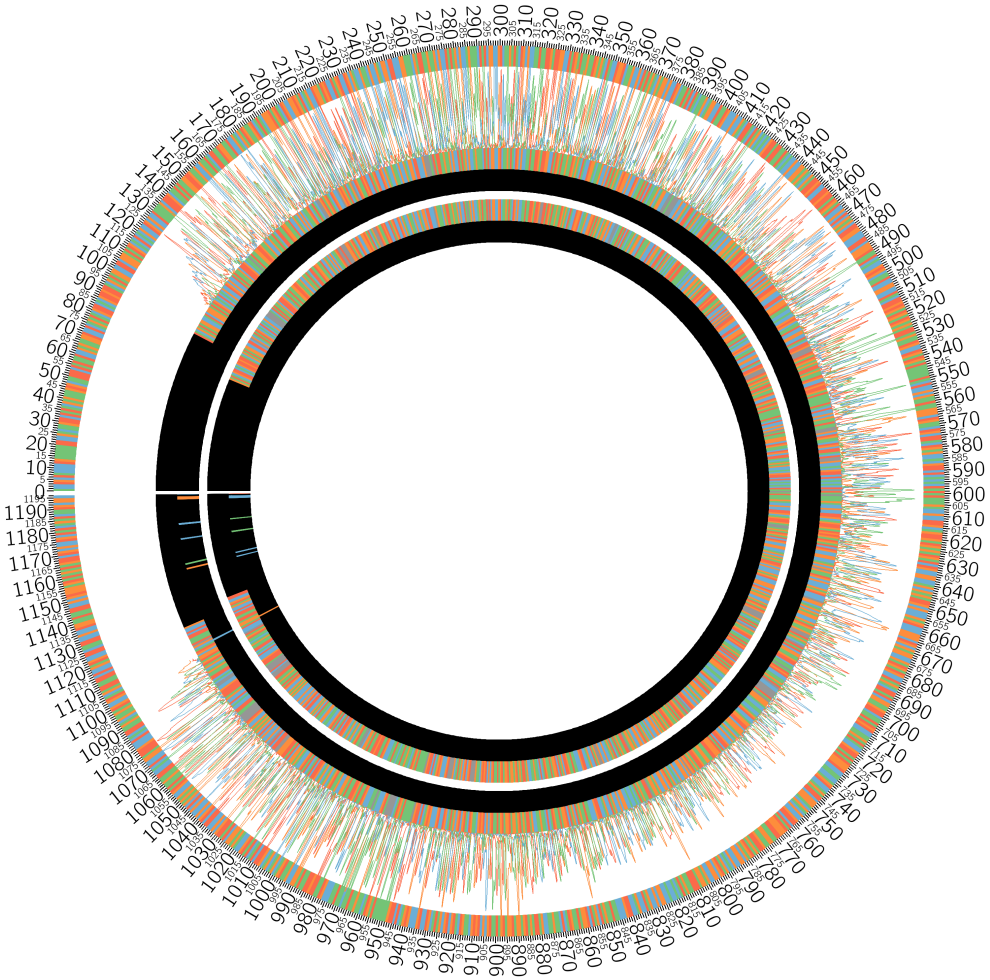


Figure C.3 `Circos` plot demonstrating `Gretel` haplotype recovery results for G251. `Gretel` recovered two sequences with reasonable likelihoods. Both sequences share very few variants (as evidenced by the masking of the majority of their sequences) and both have very high identity to the original reference. As `Gretel` requires variation to occur across the gene of interest, this implies significant noise in the underlying Illumina data. The actual predicted variation, save one SNP unfortunately falls outside the template captured by primer generation (positions 1155−1195). We recover two haplotypes with a single SNP difference. It should be noted that `Gretel` does not use the pseudo-reference for recovery (only as a means to align reads to a common reference), sequences are recovered from the evidence in the `Hansel` matrix. Recovering the reference from noise is non-trivial.

Colophon

This thesis was submitted in candidature for a Doctorate of Philosophy in Computational Biology (C1250S) at Aberystwyth University, by Samuel Nicholls, on Monday, 30th July 2018.

The Viva Copy was examined *viva voce* for approximately four hours, on Friday, June 8th 2018, by Prof. James McInerney (University of Nottingham) and Dr. Martin Swain (Aberystwyth University). The work was approved by the Examining Board, subject to minor corrections. This version is the Final Copy, submitted in response to the *viva voce* examination, in which the agreed amendments have been implemented and accepted by the internal examiner. This copy supersedes the Viva Copy thesis titled "*A formal definition, implementation and in vitro demonstration of recovering haplotypes from metahaplomes in natural microbial communities*" submitted on Friday, 18th May 2018.

The associated `git` commit hash for the Final Copy is `1b58e5f`. This book was printed in the Department of Computer Science Coffee Room, Aberystwyth University and bound at the Hugh Owen Library. Excluding captions, tables, listings, front and back matter, this thesis has 58452 words.

This thesis was typeset in LaTeX, with a modified version of Krishna Kumar's thesis template for Cambridge University Engineering Department. The template is open source, and available via `https://github.com/kks32/phd-thesis-template`.

`Hansel`, `Gretel` and the associated test data are available via `https://github.com/samstudio8`.

Flow diagrams were produced with `draw.io`. The majority of graphs were produced with either `Circos` or using `ggplot2`. Explanatory figures were drawn with Adobe Illustrator by the author. Photographs were provided by the author.